GEE! Grammar Error Explanation with Large Language Models

Anonymous ACL submission

Abstract

Existing grammatical error correction tools do not provide *natural language* explanations of the errors that they correct in user-written text. However, such explanations are essential for helping users learn the language by gaining a deeper understanding of its grammatical rules (DeKeyser, 2003; Ellis et al., 2006).

To address this gap, we propose the task of *grammar error explanation*, where a system needs to provide one-sentence explanations for each grammatical error in a pair of erroneous and corrected sentences. The task is not easily solved by prompting LLMs: we find that, using one-shot prompting, GPT-4 only *correctly* explains 40.6% of the errors and does not even attempt to explain 39.8% of the errors.

Since LLMs struggle to identify grammar errors, we develop a two-step pipeline that leverages fine-tuned and prompted large language models to perform structured atomic token edit extraction, followed by prompting GPT-4 to explain each edit. We evaluate our pipeline on German and Chinese grammar error correction data. Our atomic edit extraction achieves an F1 of 0.93 on German and 0.91 on Chinese. Human evaluation of generated explanations reveals that 93.9% of German errors and 96.4% of Chinese errors are correctly detected and explained. To encourage further research in this area, we will open-source our data and code.[1]

1 Introduction

Grammatical error correction (GEC) is a practical and valuable application of natural language processing that facilitates both proofreading of text and language learning. Recent advances in large language models (LLMs) have significantly improved the capabilities of GEC systems (Wang et al., 2021; Bryant et al., 2023); however, they are unable to *explain errors in natural language* alongside providing correction. Error explanation is crucial to language learning and teaching (Ellis, 2010): while corrections are a form of implicit feedback, they are not as impactful as explicit feedback (DeKeyser, 2003; Ellis et al., 2006), which involves pointing out errors and providing meta-linguistic information to the user (e.g., rules of writing well-formed phrases or sentences).

In this work, we propose a new task—*grammar error explanation* (GEE)—for which a model must generate natural language error explanations that help language learners acquire and enhance grammar knowledge. As shown in Figure 1, given a pair of sentences in which one sentence has grammar errors and the other one is corrected, a model needs

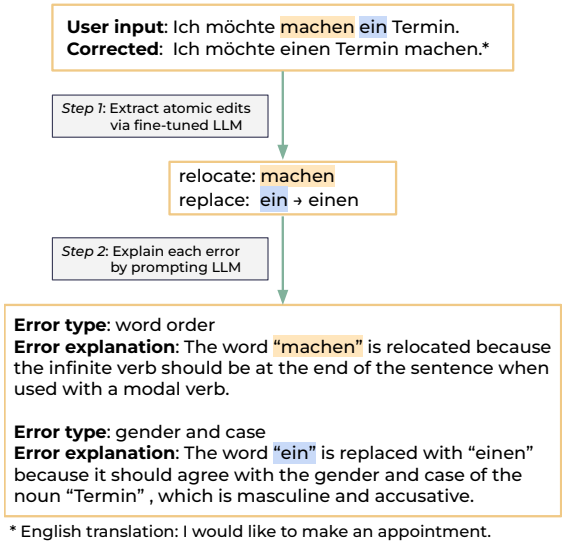

Figure 1: An illustration of the two-step pipeline of grammar error explanation (GEE). Given a pair of sentences with corrected errors, the GEE system first extracts linguistically meaningful edit units as errors. The extracted errors are then paired with the sentences as the input for GEE generation. Note: The error in *einen* can be caused by *gender* or *case* or both. Without guessing the mental state of a language user, both are offered as the reasons in the explanation.

[1] Prompts and human annotations will be made publicly available.

to generate an explanation for each corrected grammar error. Given the capabilities of modern LLMs, one might ask whether LLMs can solve this task simply via prompting. We show in Section 3 that one-shot GPT-4 (OpenAI, 2023) prompting detects only 60.2% of the true errors and correctly explains only 67.5% of the errors it does detect.

Given this result, we develop a pipeline for GEE generation that features an essential intermediate step—atomic token edit extraction. As shown in Figure 1, given an erroneous sentence and its corrected counterpart (source and target), we first extract atomic edits at the token level by prompting or fine-tuning LLMs such as GPT-4, which also label the edits with one of four operation-level edit types: `insert`, `delete`, `replace`, and `relocate`.[2] In the second step, we append the extracted edits to the source and target sentences and use them as the input to a GEE system. We utilize the few-shot learning ability of LLMs (Brown et al., 2020) to generate error explanations using carefully crafted language-specific prompts.

We validate our GEE pipeline on German and Chinese, two very different languages (fusional vs. analytical). We also recruit language teachers to evaluate the correctness of the explanations. For the first step in the pipeline, our atomic edit method extracts 92.3% of the true edits for German, which is 32.1% higher than the one-shot approach in Section 3. For the final GEE outputs in German, 93.9% of the generated explanations are judged as correct by two German teachers. Similar performance is observed in Chinese with a 96.4% correctness rate, suggesting that our two-step pipeline together with carefully crafted language-specific prompts generalizes well for the two different languages.

In summary, our contributions are the following. First, we propose a new task on grammar error explanation to enhance the utility of current grammatical error correction systems. Second, we propose a two-step pipeline and study its performance in German and Chinese with detailed error analysis. Third, we publicly release our atomic edit extraction datasets for German and Chinese as well as all LLM-generated GEE outputs with the goal of enabling future research on GEE and facilitating the development of more effective GEE systems.

[2]These types describe a general relationship between the source and target rather than precise edit operation of the source.

2 GEE task definition

While most GEC models provide viable grammar error corrections (Bryant and Ng, 2015; Bryant et al., 2023), they do not provide natural language explanations alongside the corrections, which are critical for language learners in mastering grammar (Ellis et al., 2006; Ellis, 2010). In this section, we propose and define the task of grammar error explanation, which aims to fill this gap. We assume that a GEE model has access to the outputs of an existing GEC model, which produces the corrected form of an ungrammatical input sentence.

2.1 Formalizing the GEE task

The input to a GEE model is a pair of sentences[3] in which one has (potentially multiple) grammar errors and the other is corrected. Concretely, let X_{error} be a sentence written by a user which contains grammatical errors. Then, $X_{correct} = GEC(X_{error})$ is the grammatically correct version of X_{error} produced by a GEC system. Following common practice in GEC research (Bryant et al., 2017; Lee et al., 2018; Rao et al., 2020), we assume that an error can be corrected in four ways: `insert`, `delete`, `replace`, and `relocate`. Let $c_1^X, c_2^X, ..., c_n^X$ be a list of corrections made by the GEC system to X_{error} through one of these four types of edits. Then, the goal of GEE is to generate single-sentence explanations in natural language $s_1^X, s_2^X, ..., s_n^X$ corresponding to each of $c_1^X, c_2^X, ..., c_n^X$ (example in Figure 1). Concretely,

Input: $X_{error}, X_{correct}$
Output: $s_1^X, s_2^X, ..., s_n^X$

2.2 Atomic edits as foundation of GEE

The quality of error explanation depends on how the correction list $c_1^X, c_2^X, ..., c_n^X$ is defined. Consider the corrections in (1). One way to define the correction list is through a string-based transformation (i.e., replace *machen ein termin* with *einen Termin machen*). However, an instructor explaining the corrections would naturally break them down into smaller units to facilitate understanding, for example, "*machen* must be moved to the end", "*ein* should be changed to *einen* to match gender and case", and so on. On the other hand, for the corrections in (2), an instructor would naturally explain the change as a single edit involving the movement of a phrase; breaking down the explanation

[3]In principle, the inputs could also be documents, but we restrict our work to sentence-level GEE.

into multiple word movements would not help the writer to understand why the edit was made.

(1) **S**: Ich möchte machen ein termin .
T: Ich möchte einen Termin machen .

(2) **S**: I with my puppy go to the store.
T: I go to the store with my puppy .

When explaining a corrected sentence, we argue, experts will identify the smallest individual errors that are linguistically meaningful (i.e., "atomic errors") and provide roughly one explanation per atomic error. Doing so allows learners to follow and understand explanations better, especially when there are contiguous errors in the input. This requires a process of atomic error extraction, such as the one described intuitively for (1) and (2), which naturally uses the conventions of grammar, spelling, and language usage.

We treat each atomic error as an atomic edit and give a working definition of how to identify it. Using (1) as an example, an edit (*machen ein termin*) should be divided into smaller edits (*machen*, *ein*, and *termin*) if an expert would explain the whole edit as merely the concatenation of explanations for the smaller edits. These smaller edits are then atomic edits (i.e., each of which has its own distinct explanation). Similarly, if an expert would explain an edit with multiple words using one explanation that cannot be separated into the concatenation of several explanations, then that multi-word operation is one atomic edit (e.g., the relocation of *with my puppy* in (2)).

Our working definition of atomic edits provides guidance for extracting linguistically meaningful edits. However, language-specific decisions are needed for individual languages. We discuss such details for German and Chinese in Section 4 and Appendix C.

2.3 Evaluation of GEE

We evaluate two critical aspects of GEE: error coverage and explanation quality.

Error coverage evaluation can be facilitated by forcing a model to generate position information of explained errors or to describe the edits being done. The evaluation is conducted by measuring (1) whether an explained error is indeed an error in the source and being corrected in the target; and (2) whether an error that is corrected in the target has an associated correct explanation.[4] An automatic evaluation through string overlap can give a quick estimate of error coverage when gold references are available. We also do manual evaluation to better understand the behavior of models.

Explanation quality evaluation is challenging because errors can be explained in multiple ways. To reliably evaluate GEE outputs automatically, multi-reference metrics such as METEOR (Banerjee and Lavie, 2005) and benchmarks with multiple references for each error are needed. However, collecting such datasets is costly and requires expertise in second language teaching. Without such datasets being available, leveraging human experts is the only reliable way to evaluate. In our work, we recruit language teachers for the evaluation described in Section 6.2. Language teachers, with their expertise in second language teaching, can reliably judge whether an explanation is correct and informative.

3 Has GPT-4 already solved GEE?

A natural question one might ask is whether state-of-the-art LLMs can solve the GEE task in an end-to-end manner. This section demonstrates that GPT-4 in its current form is error-prone. It has low error coverage and hallucinates frequently. Based on this observation, we experiment with an approach which provides GPT-4 a with a list of manually-extracted gold atomic edits. Results show that access to this gold edit list improves the performance greatly, indicating substantial headroom with more structured prompting as we describe in Section 4.

One-shot prompting of GPT-4. We run an experiment using German grammar error correction data (details in Section 5). We randomly sample 30 data points[5] and generate explanations using the one-shot prompt in Appendix A.

GPT-4 struggles to identify and explain errors. To get a better sense of what mistakes GPT-4 makes, we manually classify each edit as true positive, false positive, or false negative, depending on whether they are correctly mentioned in the generated explanations.[6] Our evaluation shows that GPT-4 mentioned 120 edits (compared to 90 manually extracted edits), achieving recall 0.602, precision 0.675 and F1 0.636. Among the correctly de-

[4] A GEE model should be able to ignore errors in the source sentence that are not corrected in the target sentence since GEC is not its primary task.

[5] Five data points from each CEFR level. Details are in Section 5.

[6] *Correctly mentioned* means whether one can arrive from the source to the target through the edits.

tected errors, GPT-4 only correctly explains 67.5% of them. Such outputs are not useful for learners.

GPT-4 makes mistakes even in short sentences. As the example in Table 1 shows, GPT-4 hallucinates that *Sie* is relocated. It also groups two errors together and omit the explanation of one error. The unorganized output format makes it hard to track the errors and follow the explanations.

Source: Bitte antworten sreiben Sie? **Target**: Bitte antworten und schreiben Sie.
The word 'sreiben' is replaced by 'schreiben' because there was a spelling mistake in the word. **Error type**: spelling
The word 'Sie' is relocated after 'antworten' and the word 'und' is inserted between 'antworten' and 'schreiben' because these are separate actions and should be connected with a conjunction. **Error type**: word order and conjunction

Table 1: An example of GPT-4's GEE output with the one-shot prompt. The second explanation hallucinates that 'Sie' is relocated and does not give an explanation why it is relocated. The explanation also groups the relocation of 'Sie' with the insertion of 'und'.

What if GPT-4 was provided with gold edits in the prompt? To measure the headroom for improvement, we prompt GPT-4 in the same way but provide gold atomic edits extracted manually in the input prompt. As a results, the F1 of the errors coverage is increased to 0.968. Also, 82% of the true errors receive an appropriate explanation. Hence, offering a good atomic edit list to GPT-4 is an important intermediate step. This observation motivates our proposed pipeline in Section 4, where we augment GPT-4 prompts with automatically extracted atomic edits.

4 Pipeline for generating GEE

In Section 3, we observed that including a gold list of atomic edits to GPT-4's prompt greatly improves error coverage. We thus propose a two-step pipeline for GEE that uses atomic edit extraction as the intermediate step. The pipeline is illustrated in Figure 1. Given an input sentence pair defined in Section 2.1, we first extract atomic edits from the pair following Section 2.2. The edits are then appended to the sentences to form the input for the final step, where GPT-4 is prompted to generate an explanation and an error type.

4.1 Atomic edit extraction

As discussed in Section 2.2, we define an atomic edit as the smallest individual modification that requires one explanation. Each edit belongs to one of the four operation types: `replace`, `insert`, `delete`, and `relocate`.

Previous work on edit extraction. The ERRANT system of Bryant et al. (2017) approaches edit extraction via a linguistic rule-based approach, but it has its limitations. For example, ERRANT does not account for relocated words.[7] It is also only designed for English. Adapting it to other languages requires great effort (Korre et al., 2021; Uz and Eryiğit, 2023). Further limitations of ERRANT are discussed in Appendix B. As such, we decide to use LLMs for atomic edit extraction.

Desired LLM output format. To facilitate the evaluation of edit extraction and (later) GEE generation, we restrict atomic edit extraction outputs to a template `[operation type, original token(s), target token(s)]`. An example with all four edit types is given in (3).

(3) möchte **machen ein** Termine.**?**
Ich möchte **einen** Termine **machen**.
`[insert, , Ich]`
`[relocate, machen, machen]`
`[replace, ein, einen]`
`[delete, ?, ]`

While being useful for GEE, the edit type `relocate` occasionally reduces the model performance because models tends to label a relocated token as deletion plus insertion. Relocation can also be challenging for human to decide because a relocated word should be a word order error but have the same dependency in a sentence before and after relocation. We discuss details in Appendix C.

Atomic edit extraction with LLMs. To build an atomic edit extractor, we choose to prompt Claude-2,[8] GPT-3.5-turbo-0613, and GPT-4 (via Azure's 2023-03-15-preview), as well as fine-tune Llama2-7B and GPT-3.5-turbo. For prompting, we use the carefully designed few-shot prompts in Appendix D for German and Chinese. For fine-tuning, we use Llama2-7B and GPT-3.5-turbo as the base models. We noticed that the models have a low

[7] It does account for local transposition (e.g., *juice apple* vs. *apple juice*).

[8] Accessed in November 2023. `anthropic.com/index/introducing-claude`

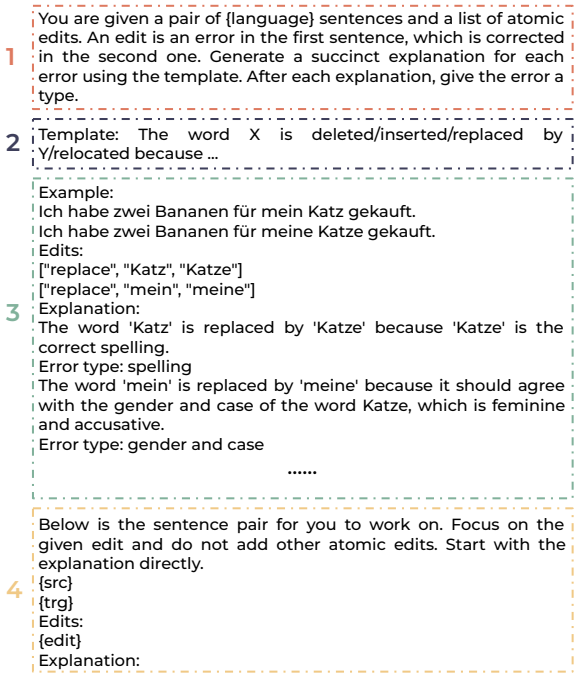

Figure 2: The prompt used for generating German grammar error explanation given an input defined in Section 2. The prompt consists of: (1) task description, (2) generic explanation template, (3) few-shot examples, and (4) current input. The full prompts for German and Chinese are in Appendix E.

recall when only sentence pairs are provided. To improve on that, we split sentences into a list of tokens and extract rough string-based edits which are the longest contiguous matching subsequences.[9] These rough edits are appended to sentence pairs as inputs. For all models, prompted or fine-tuned, we set temperature to 0 because the task does not require creativity.

4.2 GEE generation

Having extracted atomic edits, we are now ready to generate GEE. Given that each sentence pair may contain multiple errors, we investigated whether generating explanations for one error at a time or all explanations simultaneously would yield better results. In the prompt designing stage, we observed no significant difference in performance between the two approaches. Hence, we choose the latter strategy as it is efficient and cost-effective.

Figure 2 gives a shortened example of the German GEE prompt. Edits are incorporated into the input to provide context and guidance for the model. The full prompts for German and Chinese are in Appendix E. The prompts consists of four parts. The first part is the **task description**, which is followed by a generic **template** of explanations. Below the template are few-shot **examples**. In the examples, we aim to offer both meta-linguistic and meaning-oriented explanations whenever it is possible as they help L2 users improve their language skills (i.e., using languages accurately and fluently) (Lyster and Saito, 2010). At the end of the prompt, we provide GPT-4 the **sentence pair with a list of atomic edits** and ask the model to generate one explanation with an error type for each edit. The generated outputs have the following format:

```
[edit description] because [edit reason]
Error type: [error type]
```

The edit description describes how a word in the source sentence is edited in the target sentence. The edit reason explains why such an edit is made.

5 German and Chinese datasets

This section introduces the datasets that are used in our experiments. Statistics of the sampled data subsets are reported in Table 2.

	German		Chinese	
	# of data	# of edits	# of data	# of edits
Fine-tune	500	1598	496	790
Test	50	186	53	94
GEE	1122	–	970	–

Table 2: Number of sentence pairs and gold edits in each data subset in German and Chinese. We do not manually annotate the data for GEE, hence no gold edit count is reported.

5.1 German Merlin and Falko

For German GEE, we use the data from the German L2 learner corpora Falko EssayL1v2.3 (Ludeling et al., 2008; Reznicek et al., 2010) and Merlin (Boyd et al., 2014). Both datasets consist of essays written by German users whose proficiency ranges from beginners to advanced users. The datasets provide corrections of errors. The datasets are preprocessed as described in Appendix F.1.

From the preprocessed dataset, we sample two subsets without overlaps between them. First, we sample 550 data points and manually annotate them for gold atomic edits. The 550 data points are split into 500 for fine-tuning and 50 for testing, each containing 1598 and 186 gold edits. Second, for GEE generation, we sample all A1 data points (146) and randomly sample 200 data points from other CEFR levels (A2–C2). We manually remove sentence pairs that are misaligned. At the end, we obtain 1122 sentence pairs in German for GEE.

[9] We use Spacy for German and Jieba for Chinese.

5.2 Chinese CGED2017

We conduct the Chinese GEE experiment on the training split of Chinese Grammatical Error Diagnosis (CGED) 2017 (Rao et al., 2020), which are from the writing task of the *Hanyu Shuiping Kaoshi* (Test of Chinese Level) (Cui and Zhang, 2011; Zhang and Cui, 2013). Error corrections are provided but there is no learner proficiency level information. Data are preprocessed as in Appendix F.2. We sampled 520 and 60 data points for fine-tuning/prompting edit extraction models and testing performance respectively. We sample another 970 data points for generating error explanations. After cleaning, we have 496 data points for fine-tuning, 53 for testing, and 970 for explanation generation. Edit counts are in Table 2.

6 Experimental results

This section presents the results of the GEE pipeline in German and Chinese. We first present the results of the fine-tuned and prompted models on atomic edit extraction in Table 3 and 4. We find that the fine-tuned GPT-3.5-turbo achieved the best performance on edit extraction for German but GPT-4 works the best for Chinese. Section 6.2 presents the human evaluation results of German and Chinese GEE outputs generated by GPT-4. Among the German GEE outputs, 93.9% are judged as correct by two German teachers. For Chinese GEE outputs, 96.4% of the outputs are correct according to two Chinese teachers.

6.1 Atomic edit extraction results

We first describe our experimental setup before diving into the performance of fine-tuned and prompted models. Results are presented in Tables 3 and 4 for German and Chinese, respectively.

Experiment setup. We few-shot prompt Claude-2, GPT-3.5-turbo, and GPT-4 with the prompt for German in Appendix D.1. For fine-tuning, we use Llama2-7B and GPT-3.5-turbo as the base models and fine-tune them on the 500 training data points in Table 2. Details of the fine-tuning process are in Appendix G. At inference time, the temperature of all models is set to 0. We employ simple heuristics to post-process model outputs to remove low-level false positive errors, such as replacement edits that have the same original and edited tokens.

Evaluation. While automatic evaluation is fast, we evaluate the test data manually because there can be multiple ways to get to a target sentence from a source sentence. Concretely, we compare model edits against the manually extracted gold edits one by one. When there is a discrepancy, if the model outputs are linguistically meaningful and can reach the same target, we treat them as true positives.

	Claude-2	Llama2-7B	GPT-3.5-turbo		GPT-4
	Prompting	Fine-Tuned	Prompting	Fine-Tuned	Prompting
Recall	0.789	0.849	0.695	**0.923**	0.874
Precision	0.737	0.827	0.764	**0.939**	0.870
F1	0.762	0.838	0.728	**0.931**	0.870
Edit Count	199	191	161	180	184

Table 3: Recall, precision, and F1 scores of models on the German atomic edit extraction task. Because of the variance in GPT-4 outputs, the outputs are generated three times and the best performance is reported.

	Claude-2	Llama2-7B	GPT-3.5-turbo		GPT-4
	Prompting	Fine-Tuned	Prompting	Fine-Tuned	Prompting
Recall	0.872	0.840	0.763	0.830	**0.884**
Precision	0.820	0.908	0.651	0.918	**0.933**
F1	0.845	0.873	0.703	0.872	**0.908**
Edit Count	100	87	109	85	90

Table 4: Recall, precision, and F1 scores of models in the Chinese atomic edit extraction task. Because of the variance in GPT-4 outputs, the outputs are generated three times and the best performance is reported.

Results on German: fine-tuned GPT-3.5 is most effective at atomic edit extraction. The results for German edit extraction in terms of precision, recall, and F1 are in Table 3. All models have reasonable performance but the fine-tuned GPT-3.5-turbo outperforms all others. It achieves 0.923 in recall, 0.939 in precision, and 0.931 in F1. We use it as the atomic edit extractor in the next step in German GEE generation.

Results on Chinese: prompted GPT-4 is the most effective edit extractor. The results are reported in Table 4. Unlike German, the prompted GPT-4 returns the best performance. Because of the variance in the GPT-4 outputs, we verify its performance by running the experiment three times. All three runs of GPT-4 return the highest scores. The best results of GPT-4 are recall 0.884, precision 0.933, and F1 score 0.908. We hypothesize that the reason of the prompted GPT-4 performing well on Chinese is that each Chinese sentence pair has less edits on average (see Table 2). The same reason leads to the fact that there are less edits in the training data, which might cause the fine-tuned models perform worse than the ones in German.

6.2 Human evaluation of GEE

To evaluate the performance of our GEE pipeline, we recruited two German teachers and two Chinese

teachers.[10] This section provides quantitative results from the human evaluations of GPT-4 on the generated GEEs for German and Chinese. Detailed qualitative analysis is in Appendix I.

The results indicate that our GEE pipeline generates explanations of which 93.9% and 96.4% are correct for German and Chinese, respectively. However, we find that GPT-4 occasionally produces low-level errors such as formatting issues. For Chinese, when it comes to word choice errors, GPT-4 does not always provide clear contrast between two words. It also produces overly general error types.

6.2.1 Human evaluation of German GEE

German GEE generation. Using the best performing edit extractor from Section 6.1, we extract atomic edits from the 1122 sentence pairs described in Section 5. The extracted edits are paired with the source and target sentences to prompt GPT-4 using the few-shot prompt in Appendix E.1. We use the default hyperparameters offered by the OpenAI API (i.e., temperature $= 1$ and top p $= 1$) for some creativity in the explanations.

German GEE evaluation setting. The annotation interface is shown in Figure 4. We collected annotations on error explanations of 596 unique German sentence pairs. To assess the agreement between the teachers, 96 pairs are annotated by both of them. A total of 692 sentence pairs were annotated for this study.[11] The two teachers' agreement rate is 89.6%. Details of the agreement assessment and evaluation instructions are in Appendix H.

Human annotation protocol for evaluating GEE. For each sentence pair, we present the explanations generated by GPT-4 to the teachers, who are asked to check for four types of mistakes:[12]

- **Hallucinated error**: an error in an explanation that does not exist in the source sentence. Such a mistake can be made by considering a correct word/punctuation as an error, or it can be a word that does not exist in the sentences at all.
- **Missing error**: an error in the source which is edited in the target but not explained.
- **Wrong error explanation**: wrong edit description, wrong edit reason, or both.
- **Wrong error type**: an error type that is not related to the explained error.

	Count	Percentage
Fully correct	1865	93.9%
Wrong error explanation	94	4.7%
Wrong error type	12	0.6%
Hallucinated error	15	0.8%
Total explanation count	1986	100%
Total annotated items	692	
Missing error	67	

Table 5: Results of human evaluation on German GEE by two German teachers. 692 sentence pairs with 1986 explanations are annotated. GPT-4 generates fully correct edit description, edit reason, and error type 93.9% of the time. There are 4.7% *wrong error explanation* mistakes. The count of *missing errors* by the teachers is the lower bound of the actual ones.

German GEE using edit-driven GPT-4 prompts has high quality. The counts of each mistake type are reported in Table 5. The results show that GPT-4 generates correct explanations 93.9% of the time. The occurrences of inappropriate error types and hallucinated errors are both below 1%. Among the 94 *wrong error explanations*, 65 are wrong in the edit description but correct in edit reason. Among those 65 edit description mistakes, as many as 31 are because GPT-4 describes inserted and deleted edits as *The word '' is inserted/deleted because ...* without mentioning the word itself. Among the 15 hallucinated errors, 12 are caused by wrong atomic edit extraction and 3 are hallucinated by GPT-4 in the process of generating explanations.

Remaining issues. To gain a deeper understanding of GPT-4's limitations, we look into its mistakes in detail and notice that GPT-4 does not consider a context that is sufficiently large for certain errors, especially when it comes to prepositions. For example, when explaining the error in *mit 2 Zimmer* vs. *mit 2 Zimmern*, GPT-4 only says that the dative case is needed here. It does not consider the close-by preposition *mit* which requires a dative case of its complement. We provide a detailed analysis of other errors in the GPT-4 outputs in Appendix I.

6.2.2 Human evaluation of Chinese GEE

To understand how generalizable our pipeline is to different types of languages, we evaluate its performance on Chinese using the CGED2017 data

[10] Both German teachers give classes 15 to 20 hours per week. One Chinese teachers teaches 4 classes a week and the other 22-28 hour a week.

[11] There are 2082 edits extracted from 692 sentence pairs, but GPT-4 only generates explanations for 1986 of them.

[12] We call grammar errors in sentences as errors and errors made by GPT-4 as mistakes.

described in Section 5. Two Chinese teachers evaluated Chinese GEE outputs on 356 sentence pairs with 523 explanations.[13] The annotation task is set up in the same way as German. The agreement rate is 92.9% (see Appendix H).

	Count	Percentage
Fully correct	504	96.37%
Wrong error explanation	10	1.91%
Wrong error type	9	1.72%
Hallucinated error	0	0.0%
Total explanation count	523	100%
Total annotated items	356	
Missing error	1	

Table 6: Results of human evaluation on Chinese GEE by two Chinese teachers. 96.37% of the generated explanations are judged as correct. 356 sentence pairs with 523 explanations are annotated. The evaluation criteria are the same as for German.

Positive findings. Among the 356 annotated explanations, 96.37% are judged as correct by the Chinese teachers. GPT-4 has low mistake rates in all four mistake types. This shows that the proposed pipeline is effective and adaptable for very different languages like German and Chinese.

Remaining issues. While GPT-4 achieves high correctness rate in Chinese GEE, there are three caveats. First, during the data annotation for gold atomic edits, we notice that most of the edits are simple and can be readily extracted by a string-based tool. The reason is that each sentence pair on average has fewer edits than in the German data (see Table 2). Second, GPT-4 often generate generic error types. For example, it considers *idiomatic expression* errors as simply *word choice* errors. Third, for true word choice errors, GPT-4 does not always give a clear comparison of word meanings. For example, in (4), GPT-4 only explains what 严重 (serious) means but not why 严重的问题 (serious problem) is good but 严重性的问题 (seriousness problem) is not.

(4) 严重性的问题 → 严重的问题
The word '严重性' is replaced with '严重' because '严重' is the correct word for 'serious' when describing the severity of a problem.

Because word choice is a prevalent problem in Chinese grammar errors (see Table 12 for error types generated by GPT-4), such clear comparison should be enforced in an explanation so that language learners can draw inferences about other cases from the current error.

[13] There are 543 edits extracted from the 356 sentence pairs. GPT-4 only generates explanations for 523 of them.

7 Related work

This section reviews related work on GEC and feedback generation. Details are in Appendix J.

Bryant and Ng (2015); Zhang et al. (2022); Xu et al. (2022) investigate GEC with multi-reference. In GEE, a capable system should generate well-suited explanations for any valid error corrections, which requires reasoning of word relations and recovering correction rationales. Such ability also need to go beyond the sentence level. Wang et al. (2022) has shown that even when only one sentence is used as the context, a GEC model's performance can be significantly boosted. If some errors can only be better corrected in context, they can be better explained in context as well. Fei et al. (2023) find that adding evidence words into a GEC model significantly increases its performance on English GEC. For the GEE task, it is an interesting direction to explore whether adding those extra information can improve its explanations' usefulness.

On the side of explanation generation, Nagata et al. (2021) proposed a shared task called *feedback comment generation for language learners* (FCG). The task differs from our GEE task in three important aspects. First, the inputs in FCG are erroneous sentences only, which have spans marked as errors. Hence, the FCG task does not need to extract meaningful atomic edits. Second, the FCG task focuses solely on preposition words, which are a closed set of function words whose occurrences and usages are limited. Third, the FCG task focuses on generating comments as hints for language learners to correct errors themselves.

8 Conclusion

We present a new task grammar error explanation to provide natural language explanations to grammatical errors. We develop a pipelined approach using LLMs and atomic token edit extraction. Our LLM-based pipeline gets a high score of 93.9% in German and 96.37% in Chinese error explanation.

While we assume a grammar error correction system as the foundation of our GEE system, further work are encouraged to explore GEE generation alongside GEC.

Limitations

We acknowledge two limitations of our current work. First, our grammar error explanation system only considers sentence level inputs. However, certain error types (e.g., word choice and coreference) can benefit from a larger context. Second, because the Chinese data used in our work are from the HSK test (Test of Chinese Level), the covered topics are limited. It also does not include data from learners from all proficiency levels. Hence, the error types might not be representative for all levels of Chinese learners.

Ethical Considerations

Overall, our project had a small computational cost since we used QLoRA (Dettmers et al., 2023) for efficient model fine-tuning on one RTX8000. Although we do not know how GPT-3.5-turbo fine-tuning is done, each round of GPT-3.5-turbo fine-tuning took about 30 minutes. All fine-tuning and inference experiments in this paper can be completed within a day.

For the annotation work, we estimated that each annotated item on average would take one minute. As a result, we paid annotators $15 per hour. Additional bonus are paid for reasonable extra time spent on the task.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Shabnam Behzad, Amir Zeldes, and Nathan Schneider. 2023. Sentence-level feedback generation for English language learners: Does data augmentation help? In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 53–59, Prague, Czechia. Association for Computational Linguistics.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).

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, 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 and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 697–707, Beijing, China. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical Error Correction: A Survey of the State of the Art. *Computational Linguistics*, pages 1–59.

Steven Coyne. 2023. Template-guided grammatical error feedback comment generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 94–104, Dubrovnik, Croatia. Association for Computational Linguistics.

Steven Coyne, Diana Galvan-Sosa, Keisuke Sakaguchi, and Kentaro Inui. 2023. Developing a typology for language learning feedback. In *Proceedings of the 29th Annual Conference of the Association for Natural Language Processing*, Okinawa, Japan.

Xiliang Cui and Bao-lin Zhang. 2011. The principles for building the "international corpus of learner chinese". *Applied Linguistics*, 2:100–108.

Robert DeKeyser. 2003. Implicit and explicit learning. *The handbook of second language acquisition*, pages 312–348.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Rod Ellis. 2010. Epilogue: A framework for investigating oral and written corrective feedback. *Studies in second language acquisition*, 32(2):335–349.

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.

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.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Kunitaka Jimichi, Kotaro Funakoshi, and Manabu Okumura. 2023. Feedback comment generation using predicted grammatical terms. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 79–83, Prague, Czechia. Association for Computational Linguistics.

Katerina Korre, Marita Chatzipanagiotou, and John Pavlopoulos. 2021. ELERRANT: Automatic grammatical error type classification for Greek. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 708–717, Held Online. INCOMA Ltd.

Katerina Korre and John Pavlopoulos. 2020. ERRANT: Assessing and improving grammatical error type classification. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 85–89, Online. International Committee on Computational Linguistics.

Lung-Hao Lee, Yuen-Hsien Tseng, and Li-Ping Chang. 2018. Building a TOCFL learner corpus for Chinese grammatical error diagnosis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

A Ludeling, Seanna Doolittle, Hagen Hirschmann, Karin Schmidt, and Maik Walter. 2008. Das Lernerkorpus Falko. *Deutsch als Fremdsprache*, 45(2):67.

Roy Lyster and Kazuya Saito. 2010. Interactional feedback as instructional input: A synthesis of classroom SLA research. *Language, Interaction and Acquisition*, 1(2):276–297.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. Linguistic rules-based corpus generation for native Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

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

OpenAI. 2023. GPT-4 technical report.

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

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35,

Suzhou, China. Association for Computational Linguistics.

Marc Reznicek, Maik Walter, Karin Schmidt, Anke Lüdeling, Hagen Hirschmann, Cedric Krummes, and Torsten Andreas. 2010. Das Falko-Handbuch: Korpusaufbau und annotationen. *Institut für deutsche Sprache und Linguistik, Humboldt-Universität zu Berlin, Berlin*.

Maja Stahl and Henning Wachsmuth. 2023. Identifying feedback types to augment feedback comment generation. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 31–36, Prague, Czechia. Association for Computational Linguistics.

Katherine Thai, Marzena Karpinska, Kalpesh Krishna, Bill Ray, Moira Inghilleri, John Wieting, and Mohit Iyyer. 2022. Exploring document-level literary machine translation with parallel paragraphs from world literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9882–9902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Naoya Ueda and Mamoru Komachi. 2023. TMU feedback comment generation system using pretrained sequence-to-sequence language models. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 68–73, Prague, Czechia. Association for Computational Linguistics.

Harun Uz and Gülşen Eryiğit. 2023. Towards automatic grammatical error type classification for Turkish. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 134–142, Dubrovnik, Croatia. Association for Computational Linguistics.

Baoxin Wang, Xingyi Duan, Dayong Wu, Wanxiang Che, Zhigang Chen, and Guoping Hu. 2022. CCTC: A cross-sentence Chinese text correction dataset for native speakers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3331–3341, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Trans. Intell. Syst. Technol.*, 12(5).

Fei Xia. 2000. The segmentation guidelines for the penn chinese treebank 3.0. *IRCS Technical Reports Series. 37.*

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. FCGEC: Fine-grained corpus for Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lingyu Yang, Hongjia Li, Lei Li, Chengyin Xu, Shutao Xia, and Chun Yuan. 2023. LET: Leveraging error type information for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5986–5998, Toronto, Canada. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. 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.

Bao-lin Zhang and Xiliang Cui. 2013. Design concepts of "the construction and research of the interlanguage corpus of chinese from global learners". *Language Teaching and Linguistic Study*, 5:27–34.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

A One-shot prompt for GPT-4

We use the following one-shot prompt for the German experiment in Section 3 which shows that GEE cannot be solved end-to-end by GPT-4.

```
You are given a pair of German sentences. The
    first sentence contains one or more errors,
    which are corrected in the second one. Your
    task is to: (1) generate a succinct
    explanation for each error following the
    template; (2) assign the error a type.

Template: The word X is deleted/inserted/
    replaced by Y/relocated because ...

Example:
Ich habe zwei bananen für mein Katze gekauft.
Ich habe zwei Bananen für meine Katze gekauft.
Explanation:
The word 'bananen' is replaced by 'Bananen'
    because German nouns should be capitalized.
Error type: capitalization
The word 'mein' is replaced by 'meine' because
    it should agree with the gender and case of
    the word Katze, which is feminine and
    accusative.
Error type: gender and case agreement

Below is the sentence pair for you to work on.
    Start with the explanation directly.
```

```
{src}
{trg}
Explanation:
```

B Reasons of not using ERRANT

ERRANT (Bryant et al., 2017) is an effort to standardise datasets for GEC, reduce annotators' burden, and offer feedback to instructors and learners. It does so by offering a tool that automatically extracts and labels edits in the format of `operation:linguistic feature`.

ERRANT would have been ideal for our purpose. Concretely, this would have been ideal for the edit extraction in Step 1 and error type tagging in Step 2. However, ERRANT has several shortcomings for our purpose.

First, ERRANT is designed only for English and its error type tagging process is based on a English rule-based framework. Extending it to another language will take great effort (Korre et al., 2021; Uz and Eryiğit, 2023).

Second, there is ambiguity in ERRANT's error type names. For example, `R:ADV` is a possible error type in ERRANT in which `R` stands for replacement and `ADV` stands for adverb. But it is not clear, as it stands, whether it represents only an adverb being replaced by another adverb, or it could be the case that a word of other category is replaced by an adverb.

Third, Korre and Pavlopoulos (2020) show that ERRANT can falsely or ambiguously tag errors. In their work, they use ERRANT to tag the errors in the FCE dataset (Yannakoudakis et al., 2011). They then sample 100 sentence pairs to whose errors ERRANT assigned the type `Other`. They examine those sentence pairs and found that up to 39% of the data point could have been assigned a more precise label.

Fourth, ERRANT's underlying edit extractor does not account for non-local token relocation (Felice et al., 2016). The extractor aligns the tokens in the erroneous and correct sentences and assign one of the following labels to spans: `M(atch)`, `I(nsertion)`, `D(eletion)`, `S(ubstitution)`, and `T(ransposition)`. For a relatively locally relocated token, the extractor assigns the label `T` to the span as in (1). However, for a less local token relocation such as (2), the extractor treats it as being deleted then inserted.

(1) Ich_0 möchte_1 **haben**$_2$ einen_3 Apfel_4 .5

Ich_0 möchte_1 einen_2 Apfel_3 **haben**$_4$ .5

```
('M', 0, 1, 0, 1)
('M', 1, 2, 1, 2)
('T3', 2, 5, 2, 5)
('M', 5, 6, 5, 6)
```

(2) Ich_0 möchte_1 **haben**$_2$ einen_3 roten_4 Apfel_5 .6

Ich_0 möchte_1 einen_2 roten_3 Apfel_4 **haben**$_5$ .6

```
('M', 0, 1, 0, 1)
('M', 1, 2, 1, 2)
('D', 2, 3, 2, 2)
('M', 3, 4, 2, 3)
('M', 4, 5, 3, 4)
('M', 5, 6, 4, 5)
('I', 6, 6, 5, 6)
('M', 6, 7, 6, 7)
```

Relocation of tokens would be a useful label to have for word order errors, which are prevalent in elementary L2 German and Chinese learners. With this label, we could explain why a token is relocated rather than explaining why it is deleted first then explaining why it is inserted.

C Guidelines for manual edit extraction Annotation

To prepare the data for fine-tuning models to extract atomic edits in German and Mandarin Chinese, we manually annotated 500 data points for each language. In this section, we discuss the challenges in extracting atomic edits and how we handle them.

The first step is to tokenize sentences. For German, it is straightforward because of white spaces. We use `SpaCy` for tokenizing German sentences which can single out punctuation marks. For Chinese, sentences cannot be tokenized into words by simply separating characters because many words are not monosyllabic. We choose to use Jieba, which is a fast and accurate Chinese word segmentation module implemented in Python.

The second step is to use `SequenceMatcher` from `difflib` to extract longest edited spans from sentence pairs, which is later used as part of the input for atomic edits. We found that adding rough edits into the input increases the recall of the prompted models. It also accelerates and eases the process of manual annotation.

The third and last step is to get atomic edits. There are four types of edits: `replacement`,

deletion, insertion, and relocate. The challenge lies in how to align words in sentence pairs and extract edits.

For German, **replacement** mostly happens between tokens which have similar spelling (e.g., *wolle* and *will*, meaning *want to*) or the same categories (e.g., *zu* and *nach*, meaning *to*). **Deletion** and **insertion** can happen to individual tokens or a phrase. When more than one consecutive tokens, for example, X and Y, are deleted or inserted, we determine whether to count them as separate edits or one as a whole depending on whether X and Y form a linguistic constituent (for example, a prepositional phrase *by train*). The edit type **relocation** is inspired by a common error made by elementary German learners: placing finite verbs or adverbial phrases in the wrong position.[14] To emphasize that the usage of a word is not wrong but its position in a sentence is wrong, tagging such an edit as relocated is more intuitive than tagging it as a deletion followed by an insertion (or an insertion followed by a deletion).

The introduction of the relocation edit type is not at no cost. It reduces model performance because models tends to predict a relocated token/phrase as deletion plus insertion. It is also challenging because the relocated word should be just placed in a wrong position and have the same dependency in a sentence before and after being relocated. For example, for the sentences in (5), it is illogical to say that the first sentence is corrected by relocating *for* to the first underline and insert *to* in the original place of *for*. This is because the verb *talk* requires a preposition but the language user mistakenly used *for* instead of *to*. It is not the case that the language user mistakenly put the *for* that should have been before *me* after *talking*. So, it should be the case that *for* is inserted to the position of the blank underline and the *for* after *talking* is replaced by *to*. The correct edits for (5) are given in (6) and the wrong edits are in (7).

(5) **S**: This job is exciting ___ me because I like talking for different people.
T: This job is exciting for me because I like talking to different people.

(6) Good edit extraction
`['insert', '', 'for']`
`['replace', 'for', 'to']`

(7) Bad edit extraction
`['relocate', 'for', 'for']`
`['insert', '', 'to']`

On the other hand, the word *essen* in (8) is more naturally a relocated token because its relation with the finite modal verb *moöchte* (would like to) and the direct object *vierzig Bananen* (forty bananas) remains unchanged. It is only the position of the word that is changed.

(8) **S**: Ich möchte essen vierzig Bananen.
T: Ich möchte vierzig Bananen essen.
`['relocate', 'essen', 'essen']`

For Chinese, deletion and insertion work similarly as in German. Relocation is also useful in Chinese for cases like misplacement of an adverbial phrase or a function word (e.g., 了).[15] However, replacement is not as straightforward in Chinese as in German. For example, verbs in Chinese often come with a resultative complement (e.g., 到, 完, or 出) or other function words to express different states of a verb (e.g., 过). If only the function word is changed but the verb is not, how should the edit be extracted? We experimented with both ways (with and without verbs) and found that, in either case, GPT-4 included the verb when explaining the meaning difference. Hence, for those cases, we always include the unchanged verbs, as in (9). Similarly, for cases in which a function word is not changed but the verb that the function word is attached to is changed, the edit includes both the verb and the function word (e.g., `['replace', '看成', '当成']`).

(9) **S**: 我花了一整天看过了这本书。
T: 我花了一整天看完了这本书。
`['replace', '看过', '看完']`

Other situations in which we always take longer phrases as edits rather than only the parts being changed are idioms (e.g., `['replace', '心急如坟', '心急如焚']`), formulaic expressions (e.g., `['replace', '总上所述', '综上所述']`), and *de* (的)+ noun as in 在这紧急的情况下 (in an emergency situation).

[14] German is a verb second language, whose verb second constraint does not hold in embedded clauses. In main clauses, the finite verb occurs in the second position and non-finite verbs occur towards the end of a sentence. In embedded clauses, the finite verb usually appears at the end, after all the non-finite verbs.

[15] 了 is a multi-functional function word and a heteronym. It can express the completion or ongoingness of an action (among its other functions). Its meaning changes based on the position in a sentence it occurs.

D Prompts for atomic edit extraction

We use the prompts presented below for atomic edit extraction in German and Chinese. The prompt contains the task instruction followed by possible edit types as well as examples. Special instructions are given to the relocation edit type where the relocated tokens should be the same before and after the edit. In the examples, we demonstrate different edit types and their combinations, showing the models how to deal with a sentence pair with multiple edits.

D.1 Extraction prompt for German

```
This is an atomic edit extraction task. Given a
    pair of German sentences and the edits
    applied to the first sentence to get the
    second sentence, your task is to break down
    the edits to the atomic level (i.e., token
    level) and assign the edit a label. Be case
    sensitive. Pay attention to punctuation
    marks and relocated tokens. Pay attention to
     phonetic similarity when aligning tokens.

Labels:
1. [replace, original_token, edited_token]
2. [delete, original_token, ""]
3. [insert, "", edited_token]
4. [relocate, original_token, edited_token]: pay
     attention to tokens that are deleted then
    added again; the relocated token must be the
     same before and after the edit.

Examples:
Wie oben schon erwähnt ist die Chance erwisht
    zurweden zwar gering, aber sie ver handen.
Wie oben schon erwähnt ist die Chance, erwischt
    zu werden, zwar gering, aber sie ist
    vorhanden.
Edits:
('replace', 'erwisht zurweden', ', erwischt zu
    werden ,')
('replace', 'ver handen', 'ist vorhanden')
Atomic edits:
["insert", "", ","]
["replace", "erwisht", "erwischt"]
["replace", "zurweden", "zu werden"]
["insert", "", ","]
["insert", "", "ist"]
["replace", "ver handen", "vorhanden"]

ich haben essen zwei Bananen.
Ich habe zwei Bananen gegessen.
Edits:
('replace', 'ich haben essen', 'Ich habe')
('insert', '', 'gegessen')
Atomic edits:
["replace", "ich", "Ich"]
["replace", "haben", "habe"]
["delete", "essen", ""]
["insert", "", "gegessen"]

Ich habe gegessen zwei Bananen.
Ich habe zwei Bananen gegessen.
Edits:
('delete', 'gegessen', '')
('insert', '', 'gegessen')
Atomic edits:
["relocate", "gegessen", "gegessen"]

Below is the sentence pair for you to work on.
    Follow the format in the examples strictly.
{src}
{trg}
Edits:
{edits}
Atomic edits:
```

D.2 Extraction prompt for Chinese

You are a Mandarin Chinese teacher. Given a pair of Mandarin Chinese sentences and the edits applied to the input sentence to get the output sentence, your task is to break down the edits to the atomic level (i.e., token level) and assign the edit a label. Pay attention to punctuation marks and relocated tokens.
Labels:
1. [replace, original_token, editted_token]
2. [delete, original_token, ""]
3. [insert, "", editted_token]
4. [relocate, original_token1, editted_token1]: pay attention to tokens that are deleted then added again; the relocated token must be the same before and after the edit.
Examples:
我去市菜场水果买。
我去菜市场买水果。
Edits:
("replace", "市菜场水果买", "菜市场买水果")
Atomic edits:
["replace", "市菜场", "菜市场"]
["relocate", "水果", "水果"]
我吃了早饭今天。
我今天吃了早饭。
Edits:
("insert", "今天", "")
("delete", "", "今天")
Atomic edits:
["relocate", "今天", "今天"]
再子细的学习相关课题后，我意识到了这个问题的严重。
在仔细地学习了相关课题后，意识到了这个问题的严重性。
Edits:
("replace", "再子细的", "在仔细地")
("insert", "", "了")
("insert", "", "我")
("insert", "", "性")
Atomic edits:
["replace", "再", "在]

["replace", "子细", "仔细"]
["replace", "的", "地"]
["insert", "", "了"]
["insert", "", "我"]
["replace", "严重", "严重性"]
她打算明儿天的午前去北京。
她打算明天上午去北京。
Edits:
("replace", "明儿天的午前", "明天上午")
Atomic edits:
["replace", "明儿天", "明天"]
["delete", "的", ""]
["replace", "午前", "上午"]
Below is the sentence pair for you to work on. Follow the format in the examples strictly.
{original_sentence}
{corrected_sentence}
Edits:
{edits}
Atomic edits:

E Prompts for explanation generation

We use the following prompts for generating grammar error explanations in German and Chinese.

E.1 Explanation prompt for German

In the prompt for German grammar error explanation, we provide a wide range of error examples, including errors that can only be explained in grammatical terms (e.g., gender/case/number agreement), errors that can be assigned a meaning (e.g., accusative case for time expressions), and errors that are related to collocations (e.g., *am Ende* instead of *im Ende*).

```
You are given a pair of German sentences and a
    list of atomic edits. An edit is an error in
     the first sentence, which is corrected in
    the second one. Generate a succinct
    explanation for each error using the
    template. After each explanation, give the
    error a type.

Template: The word X is deleted/inserted/
    replaced by Y/relocated because ...

Example:
Ich habe zwei Bananen für mein Katz gekauft.
Ich habe zwei Bananen für meine Katze gekauft.
Edits:
["replace", "Katz", "Katze"]
["replace", "mein", "meine"]
Explanation:
The word 'Katz' is replaced by 'Katze' because '
    Katze' is the correct spelling.
Error type: spelling

The word 'mein' is replaced by 'meine' because
    it should agree with the gender and case of
    the word Katze, which is feminine and
    accusative.
Error type: gender and case

Er fliegt nächster Monat Deutschland.
Er fliegt nächsten Monat nach Deutschland.
Edits:
["insert", "", "nach"]
["replace", "nächster", "nächsten"]
Explanation:
The word 'nach' is inserted because the verb '
    fliegen' requires a preposition when
    expressing a destination and 'nach' is
    usually used for countries.
Error type: preposition
The word 'nächster' is replaced by 'nächsten'
    because German uses accusative case for time
     expressions.
Error type: case

Ich gehe in der Schule.
Ich gehe in die Schule.
Edits:
["replace", "der", "die"]
Explanation:
The word 'der' is replaced by 'die' because the
    preposition 'in' requires the accusative
    case of a noun when expressing a direction
    or destination.
Error type: case

Ich kann heute jogge gehe.
Ich kann heute joggen gehen.
Edits:
["replace", "gehe", "gehen"]
["replace", "jogge", "joggen"]
Explanation:
The word "gehe" is replaced by "gehen" because
    the verb "kann" requires an infinitive form
    of the verb "gehen".
Error type: infinitive
The word "jogge" is replaced by "joggen" because
     the verb "gehen" requires an infinitive
    form of the verb "joggen".
Error type: infinitive

Ich muss mich zur neuen Umgebung gewöhnen.
Ich muss mich an die neue Umgebung gewöhnen.
Edits:
["replace", "zur", "an"]
["insert", "", "die"]
["replace", "neuen", "neue"]
Explanation:
The word "zur" is replaced by "an" because the
    verb "gewöhnen" requires the preposition "an
    ".
Error type: preposition
The word "die" is inserted because the noun "
    Umgebung" requires a determiner and "gewö
    hnen an" requires accusative case.
Error type: determiner
The word "neuen" is replaced by "neue" because
    the existence of "die" indicates that the
    adjective need only weak inflection.
Error type: adjective inflection

Es ist im Ende des Flusses.
Es ist am Ende des Flusses.
```

```
Edits:
["replace", "im", "am"]
Explanation:
The word "im" is replaced by "am" because "am"
    is the correct preposition for the word "
    Ende".

Below is the sentence pair for you to work on.
    Focus on the given edit and do not add other
     atomic edits. Start with the explanation
    directly.
{src}
{trg}
Edits:
{edit}
Explanation:
```

E.2 Explanation generation prompt for Chinese

In the few-shot prompt for Chinese GEE, we cover the following types of errors, which are commonly seen when we manually annotate the training data for fine-tuning: **Function word errors**, such as 了, 们, 的/地/得, and measure words; **Mis-written words/phrases**,[16] such as 平果 vs. 苹果 and 市菜场 vs. 菜市场; **Word collocation errors**, such as 做错误 vs. 犯错误; **Word choice errors**, such as 查找 vs. 寻找.

Mandarin Chinese does not have abundant agreement between words in sentences as German or English. Many errors made by learners are word choice errors. For example, 查找 and 寻找 both have the core meaning of *looking for* but the former emphasizes a systematic and methodological search for data or information while the latter suggests a more intangible search with a sense of exploration. In the example of the word choice error, we show GPT-4 that it should explain the meaning of the two words and why one is better than the other in the context. Without such an example, GPT-4 returns a generic explanation "The word X is replace by Y because Y is the correct word to use in the context." which is not helpful for language learners.

Here begins the prompt:

You are given a pair of Mandarin Chinese sentences and a list atomic edits. An edit is an error in the first sentence, which is corrected in the second one. Generate a succinct explanation for each error using the template. After each explanation, give the error a type.

Template: The word X is replaced by Y/deleted/inserted/relocated because ...

Example:
昨天我买四只平果们。
昨天我买了四个苹果。
Edits:
["insert", "", "了"]
["replace", "只", "个"]
["replace", "平果", "苹果"]
["delete", "们", ""]
Explanation:
The word '了' is inserted because '了' indicate the completion of the action '买'.
Error type: usage of '了'
The word '只' is replaced with '个' because '个' is the correct measure word for '苹果'.
Error type: measure word
The word '平果' is replaced with '苹果' because '苹果' is the correct word for 'apple'.
Error type: miswritten character/word
The word '们' is deleted because '们' is only used after pronouns or human nouns to indicate plurality.
Error type: '们'
间而说之，他唱地很好。
简而言之，他唱得很好。
Edits:
["replace", "间而说之", "简而言之"]
["replace", "地", "得"]
Explanation:
The word '间而说之' is replaced with '简而言之' because '简而言之' is the correct way of writing the phrase which means 'in short' or 'in brief'.
Error type: mis-written character/word
The word '地' is replaced with '得' because '得' is the correct 'de' particle to use when it follows a verb and the word after '得' modifies the verb.
Error type: "de" particles
许多人们做了一差误。
许多人犯了一个错误。
Edits:
["replace", "许多人们", "许多人"]
["replace", "做", "犯"]
["insert", "", "个"]
["replace", "差误", "错误"]
Explanation:
The word '许多人们' is replaced with '许多人' because when a noun is preceded by a numeral, the plural marker '们' is not needed.
Error type: '们'
The word '做' is replaced with '犯' because '犯' is the correct verb to use for the noun 'mistake'.
Error type: verb-object collocation

[16] We call them as mis-written words instead of misspelling because there is no letters or spelling in Chinese writing. Such mistakes can be made by a language user who confuses characters with the same/similar pronunciation, with similar meaning, with similar strokes, or simply remembers the wrong character order in a word.

The word '个' is inserted because a measure word is needed between the numeral and the noun and '个' is the correct measure word for '错误'.
Error type: measure word
The word '差误' is replaced with '错误' because '差误' is not a word in Chinese and '错误' is the correct word for 'mistake'.
Error type: mis-written character/word
我在查找我的知音。
我在寻找我的知音。
Edits:
["replace", "查找", "寻找"]
Explanation:
The word '查找' is replaced with '寻找' because '查找' suggests a systematic and methodological search. It usually means searching for information or data. On the other hand, '寻找' suggests a more intangible search with a sense of exploration. '寻找' fits the context better.
Error type: word choice
Below is the sentence pair for you to work on. Focus on the given edit and do not add other atomic edits. Start with the explanation directly.
{src}
{trg}
Edits:
{edit}
Explanation:

F Data preprocess for German and Chinese

This section describes how the datasets in German and Chinese are preprocessed.

F.1 Preprocess German data

The Falko dataset (Ludeling et al., 2008; Reznicek et al., 2010) contains essays written by German learners whose proficiency levels range from A1 to C1 according to the Common European Framework of Reference for Languages (CEFR).[17] The Merlin dataset (Boyd et al., 2014) is a collection of essays written by advanced German speakers from different countries with both native and non-native background. We use Merlin as C2 data.

Both Falko and Merlin offer two types of grammar error corrections, target hypothesis 1 and target hypothesis 2. Target hypothesis 1 performs minimal correction at the morpho-syntactic level while target hypothesis 2 modifies semantic and pragmatic aspects (e.g., information structure or word choice) of the input text, aiming for a more advanced paraphrase-type correction. For our purpose, we use target hypothesis 1 of each corrected sentence.[18]

To prepare the datasets, we first split the paragraphs in Falko and Merlin into sentences by adapting the paragraph alignment algorithm in Thai et al. (2022) for sentence alignment. We then screened out sentence pairs that: (1) have short sentences (less that 3 tokens); (2) contain "incomp" or "unreadable" tokens; and (3) have two sentences in the source and one sentence in the target, or vice versa, that are not merged or split.

F.2 Preprocess Chinese data

The data for Chinese GEE is the training split of CGED2017 Rao et al. (2020). Texts are split into sentences at the end of sentence punctuation (e.g., periods and question marks) and aligned.

We tokenized the sentence pairs using Jieba and show the length distribution of sentences in Figure 3. Clearly, most of the data points have 2 to 50 tokens. Each token has on average 1.8 characters. The overly long sentences (over 170 tokens) exist because of the abusive use of commas.[19] For the experiment, we select sentences of length between 5 and 50 tokens. We also remove pairs with the same source and target.

G Fine-tune atomic edit extraction models

For German we use Llama2-7B and GPT-3.5-turbo as the base models and fine-tune them on the 500 training data points in Table 2. The results show that fine-tuning GPT-3.5-turbo through the OpenAI fine-tuning API with 2 epochs and using temperature $= 0$ at the inference time returns the best performance. It took around 30 mins for fine-tuning. For Llama2-7B, we fine-tune the model with QLoRA for 1000 steps using the parameters suggested in Dettmers et al. (2023) on one RTX8000. The fine-tuning takes about five hours. Checkpoints are saved every 250 steps. At the inference time, the checkpoint saved at 750 steps with

[17] The Common European Framework of Reference for Language (CEFR) is a standard for describing language ability. There are six levels: A1, A2, B1, B2, C1, and C2. C2 is the native speaker level.

[18] Examples of the target hypothesis 1 and 2 of a corrected sentence can be found in https://gucorpling.org/amir/pdf/Reznicek_et_al.pdf.

[19] As a rough reference, Chinese Treebank 9.0 (Xia, 2000) has 132076 sentences and 2084387 tokens, which amounts to roughly 16 tokens per sentence.

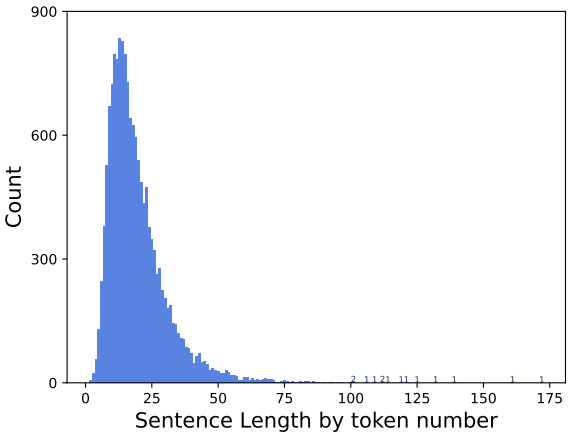

Figure 3: The sentence length distribution of the data in 2017 CGED training set Rao et al. (2020). Most of the sentences have less than 50 tokens. For the bars that are invisible in the plot, we add the numbers to them.

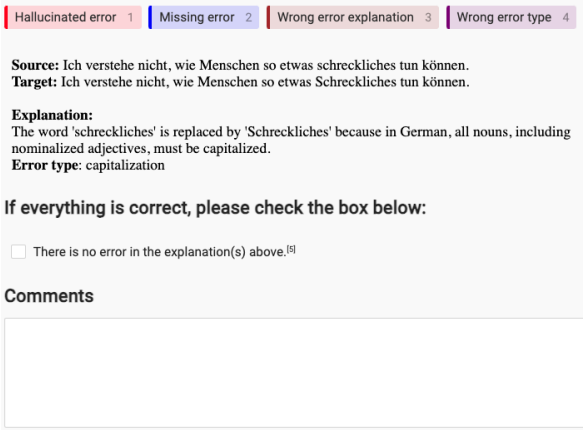

Figure 4: A screenshot of the interface presented to the annotators for explanation evaluation.

	Count	Percentage
Fully agree	78	81.3%
Disagree on missing errors	8	8.3%
Disagree on other mistakes	10	10.4%
Sum	96	100%

Table 7: Agreement between two German teachers on 96 sentence pairs. Among the 78 annotated items on which the teachers fully agree with each other, 5 have mistakes and 73 have no mistakes at all.

temperature $= 0.01$ performs the best.[20] The best performance are reported in Table 3.

For Chinese, we fine-tune Llama2-7B and GPT-3.5-turbo in the same way as for German. Llama2-7B checkpoints are saved every 100 steps. It achieves the best performance at 400 steps. Fine-tuning GPT-3.5-turbo for two epochs returns a better performance than one epoch. The best performance of the fine-tuned models are reported in Table 4.

H Details on human evaluation

We provide further details in addition to the ones discussed in Section 6.2. Figure 4 shows the annotation interface for the German and Chinese teachers. The teachers are given detailed instructions for the German (link) and Chinese (link) tasks.

In the annotation task, the teachers are asked to check for four types of mistakes. Concerning *missing error* mistakes, they should be marked either in the source sentence for deleted, replaced, and relocated tokens or in the target sentence for inserted ones. Other mistakes should be marked in the explanations. We asked the annotators not to mark imprecise explanation/error type as wrong but leave a comment on how they can be improved.

A special note on the Chinese evaluation is that, originally, each teacher annotated 200 sentence pairs, among which 100 were annotated by both. Hence, there were 400 total and 300 unique annotation items. However, there are sentence pairs whose target is judged as nonsensical or corrects errors in a wrong way. We removed those sentence pairs and report the results on the remaining items.

H.1 German annotator agreement

To evaluate the agreement, we compare the annotations of the commonly annotated 96 sentence pairs and classify them into three categories. **Fully agree**: if the teachers agree on no mistakes or the same set of mistakes. **Disagree on missing errors**: if teachers agree on other mistakes but not on *missing errors*. **Disagree on other mistakes**: if teachers also disagree on mistakes other than *missing errors*. Counts of each category are reported in Table 7.

Among the 96 commonly annotated items, the German teachers agree on 81.3% of them for the overall quality (error coverage and explanation quality), and 89.6% of the time, the teachers agree on the quality of the generated edit reasons (sum of the first and second row in Table 7).

H.2 Chinese annotator agreement

We evaluate the agreement between the two Chinese teachers on the annotation items that are anno-

[20] The `do_sample` parameter is set to `False`. The temperature is set to 0.01 instead of 0 because the model requires the temperature to strictly be a positive float.

	Count	Percentage
Fully agree	78	92.86%
Disagree on missing errors	0	0.0%
Disagree on other mistakes	6	7.14%
Sum	84	100%

Table 8: Agreement between two Chinese teachers on 84 sentence pairs. Among the 78 annotated items on which the teachers fully agree with each other, 3 have mistakes that are not *missing error* and 75 have no mistakes.

tated by both teachers. Upon inspecting the results, we notice that there are 66 sentence pairs whose target sentence has bad quality. Among them, one target sentence is nonsensical, 15 contains wrong corrections of the errors in the source sentences, and 50 of them do not correct all the errors in the source sentences.

To evaluate the agreement on the generated explanations, we remove 16 annotated items whose target is nonsensical or has wrong correction. For the remaining 84 items, we classify the annotations into the same set of categories as above. Counts of each category are reported in Table 8. Among the 84 commonly annotated items, the Chinese teachers agree on the quality of 92.86% of them.

I Qualitative analysis of German GEE

In this section, we look into the mistakes made by GPT-4 and provide detailed analysis of two of them: *wrong error type* and *wrong error explanation*.

I.1 Mistakes in wrong error type

Although there are only 12 wrong error type mistakes marked by the German teachers, they present cases where careful design decisions need to be made. We categorize them into six types and discuss two of them here. Examples and their categories are in Table 9.

Case vs. Plural The explanations and error types in the two cases indicate that, given the prompt we used, GPT-4 is weak at distinguishing certain nuances in German grammar because it does not leverage the larger context while generating explanations and error types.

In German, the suffix *-n* may occur in two cases (among others): in the plural form of certain nouns or at the end of the dative plural form of a noun if the noun's plural form does not already end in *-n*. In the first case with *Hauspreise*, the language user used *die* as the definite article of *Hauspreise*, which shows that they did not consider the case of the determiner phrase as dative. Moreover, they used *Hauspreise* as part of the subject of the sentence, which further reduces the likelihood that they meant to use *Hauspreise* in its dative case because it is very rare to have a dative determiner phrase as a subject. Hence, the error type should be *plural* or *number*. In the second case with *Menschen*, it is clearly not a *plural* error because *jede/r* (every) takes singular nouns and *bei* only takes dative nouns. The error type should be *case* because the word *Mensch* belongs to the *n*-declination which takes the *-(e)n* suffix in the dative case. Further work should add examples in the prompt or training data to enhance the model ability in distinguishing such nuances.

Misspelling vs. Conjugation While GPT-4 judges the errors under this type in Table 9 as conjugation errors, our German teachers judged them as misspelling. These three cases beg for an answer to the question: where is the border line between general misspelling due to an oversight and genuinely lack of knowledge of a grammar point (e.g., misspelling vs. conjugation)? While we do not have an answer to the question, we suggest that error types should always be the more specific one when an error is on the border line. For a language learner, if an error is made by oversight, they can easily ignore the explanation and error type. If an error is made by lacking of relevant knowledge, they should be reminded by an explanation. Since we do not know why a language learner made such an error, providing the more specific error type is more beneficial.

I.2 Mistakes in wrong error explanation

There are 29 explanations that provide a wrong reason of an error. They can be classified into two groups. The first group has mistakes that can be traced back to a wrongly extracted edit, as shown in the first example in Table 10. Eleven cases belong to this group.

The second group has mistakes for miscellaneous reasons. However, there are two reasons that stand out. The first reason is that GPT-4 does not consider information from the bigger context when generating explanations. There are 3 such cases and all of them involve a preposition. One example can be found in Table 9 under *Case vs. Plural*. Table 10 presents another one. In this example, the word *Zimmer* should be in dative not because German needs a dative case to indicate

numbers but because the preposition *mit* assigns the noun in the preposition phrase a dative case. The second reason that causes GPT-4 to generate four wrong explanations is that it does not have precise knowledge of German verb position. As in the third example in Table 10, the word *entwickelt* is relocated not because of the reason in the explanation but because a finite verb in an embedded clause should be at the end of it (see Footnote 14).

I.3 Overall quality of German GEE

In the annotation task, the German teachers were told not to mark correct but imprecise explanations/error types as wrong and leave a comment on how they can be improved. In the annotated results, we see only one such comment. That does not mean that the teachers did not leave enough comments. There are abundant comments pointing out errors in the source sentences that are not corrected in the target sentences, comments pointing out that some corrections done in the target sentences are not correct, or comments on how to modify a wrong error explanation. The first author, as a German second language learner with level C1, has also gone through all the annotated data and found the correct explanations informative and useful. Hence, we can say that the German error explanations generated by GPT-4 are judged as fully correct by our German teachers 93.9% of the time.

J More on the related work

Our GEE task is built upon the actively studied GEC task. The task is often formulated as a neural machine translation task, with the source being a piece of text with grammar errors and the target being the grammar-error-free text (Boyd, 2018; Bryant et al., 2023; Yuan and Bryant, 2021; Zhang et al., 2022). Researchers in the GEC domain have explored various aspects of the task. We identify two of them which the GEE task can be built on and benefit from. We also compare our task to a related task, feedback comment generation (FCG), and show how GEE is different from it.

GEC with multi-reference and context. Research has been building GEC models on data which have one gold reference for each source input. However, there is an urge to use multiple references for source inputs (Bryant and Ng, 2015; Zhang et al., 2022; Xu et al., 2022). In the context of GEE, a capable model should generate well-suited explanations for any valid error corrections, which requires reasoning of word relations and recovering correction rationales, not just memorize grammar rules. Such ability of GEE models also need to go beyond the sentence level. Wang et al. (2022) has shown that even when only one sentence is added to the input as the context, a GEC model's performance can be significantly boosted. If some errors can only be better corrected in context, they can only be better explained in context as well.

GEC with auxiliary grammar information. There are works that have shown improvement of GEC models by adding edit types, dependency information, or grammatical error type into the training process (Omelianchuk et al., 2020; Ma et al., 2022; Yang et al., 2023). Fei et al. (2023) study the influence of adding evidence words for errors and error types into the pipeline of GEC. They found that such information can significantly increase model performance in English GEC. For the GEE task, it is an interesting direction to explore whether adding those extra information to a GEE system can improve its explanations' usefulness.

Feedback alongside grammar error detection. On the side of explanation in GEC, Nagata et al. (2021) proposed a generative shared task called *feedback comment generation for language learners* (FCG). It is based on the ICNALE Learner Essays with Feedback Comments Dataset (Nagata et al., 2020). The task differs from our GEE task in three important aspects. First, the FCG task is built on the grammar error detection task which does not correct errors. The inputs in FCG are erroneous sentences only, which have spans marked as errors. Hence, the FCG task does not need to handle the problem of extracting errors in a linguistically informed way. Second, the FCG task focuses only on preposition words, which are a closed set of function words whose occurrences and usages are limited. In our task, the involved error types have a wide range, as listed in Table 12 for German and Table 11 for Chinese. Third, while the FCG task focuses on generating comments as hints for language learners to correct errors themselves,[21] our task aims to enhance learners' knowledge by showing them the corrected sentences, the underlying grammar rules, and a comparison with errors and corrected words when necessary.

[21] An example comment given in Nagata et al. (2021) is *"Agree" requires a preposition since it is an <intransitive verb>. Look up the appropriate preposition in a dictionary.*

Overly general
Source: Wie es deine Frau und deine Babys? Target: Wie geht es deiner Frau und deinem Baby? The word ‘’ is replaced by ‘geht’ because the correct expression in German is "Wie geht es" to ask how someone is doing. Error type: expression
Source: Brauchst du etwas hilfe? Target: Brauchst du Hilfe? The phrase ‘etwas hilfe’ is replaced by ‘Hilfe’ because ‘etwas’ is unnecessary when offering help in this context. Error type: redundancy
Source: Meine Eltren und bin ich in montag fahren in Indien. Target: Meine Eltern und ich fahren am Montag nach Indien. The phrase ‘in montag’ is deleted and replaced by ‘am Montag’ because ‘am’ is the correct preposition for expressing a specific day. Error type: preposition and expression
Misspelling vs. Conjugation
Source: Ich mochte Farkarte haben. Target: Ich möchte eine Fahrkarte haben. The word ‘mochte’ is replaced by ‘möchte’ because ‘möchte’ is the correct conjugation of the verb “mögen” in this context. Error type: conjugation
Source: Wo möchst du schwimmen gehen? Target: Wo möchtest du schwimmen gehen? The word ‘möchst’ is replaced by ‘möchtest’ because ‘möchtest’ is the correct conjugation of the verb ‘möchten’ in the second person singular. Error type: verb conjugation
Source: Man ließt es geradezu täglich in der Zeitung, hört es im Radio, oder sieht erschreckende Bilder im Fernsehn. Target: Man liest es geradezu täglich in der Zeitung, hört es im Radio, oder sieht erschreckende Bilder im Fernsehen. The word ‘ließt’ is replaced by ‘liest’ because ‘liest’ is the correct conjugation of the verb ‘lesen’ for the third person singular. Error type: conjugation
Case vs. Plural -*(e)n*
Source: In den letzten zehn Jahren erhöhten sich die Wohnungs- und Hauspreisen und fast alle müssen einen Kredit aufnehmen, ihre Träume erföllen zu können. Target: In den letzten zehn Jahren erhöhten sich die Wohnungs - und Hauspreise und fast alle müssen einen Kredit aufnehmen, um ihre Träume erföllen zu können. The word ‘Hauspreisen’ is replaced by ‘Hauspreise’ because it should be in the nominative case, which is plural without the ‘-n’ ending Error type: case
Source: Anderseits liegt die Integration jeder Mensch an sie selbst. Target: Anderseits liegt die Integration bei jedem Menschen an ihm selbst. The word ’Mensch’ is replaced by ‘Menschen’ because the plural form is needed to agree with the pronoun ‘jedem’. Error type: plural
Because of wrong atomic edits
Source: Obwohl in Stadt X schwerig ist, eine freie Wohnung zu bekommen, gibt es trotzdem viele Möglichkeiten wie man es sie bekommt. Target: Obwohl es in Stadt X schwerig ist, eine freie Wohnung zu bekommen, gibt es trotzdem viele Möglichkeiten wie man sie bekommt. The word ‘es’ is relocated because it should be placed before ‘in Stadt X’ to form the correct sentence structure for expressing a circumstance. Error type: word order
Source: Bis 30.04 ich muss in neue Wohnung umladen, weil am 01.05 meine neue Arbeit beginn. Target: Bis 30.04. muss ich in die neue Wohnung umladen, weil am 01.05. meine neue Arbeit beginnt. The word ‘ich’ is inserted because it is the pronoun that corresponds to the subject performing the action of ‘müssen’. Error type: subject pronoun
Wrong word category
Source: und werde direk zu dir für das Abholen, wäre es dir passen wenn wir den Abends zusammen Essen? Target: Und werde direkt zu dir kommen für das Abholen, würde es dir passen, wenn wir den Abend zusammen essen? The word ‘wäre’ is replaced by ‘würde’ because ‘würde’ is used in conditional sentences to express politeness. Error type: modal verb
Not understanding German phonology
Source: Hast du Seit für mich zum eine Wohnung finden? Target: Hast du Zeit, für mich eine Wohnung zu finden? The word ‘Seit’ is replaced by ‘Zeit’ because ‘Zeit’ is the correct word for time in this context. Error type: homophone

Table 9: Twelve cases in which error types are judged as wrong by the German teachers. For each sentence pair, only the explanations with a wrong error type are listed. All grammar errors in the target sentences are from the corpora.

Mistakes that can be traced back to a wrongly extracted edit

Source: Dir möchte ich nächten monat nach Amsterdam fahren.
Target: Ich möchte nächsten Monat nach Amsterdam fahren.

Atomic edits:
["replace", "Dir", "Ich"], ["replace", "ich", "nächsten"], ["replace", "nächten", "Monat"]

Explanation:
The word 'Dir' is replaced by 'Ich' because the subject of the sentence should be 'Ich' to express one's own desire to travel.
Error type: pronoun
The word 'ich' is replaced by 'nächsten' because the correct word order is 'nächsten Monat' for 'next month'.
Error type: word order
The word 'nächten' is replaced by 'Monat' because 'nächsten' should be followed by the noun 'Monat' to form the time expression 'nächsten Monat'.
Error type: word order

Mistakes due to insufficient consideration of bigger context

Source: Ich brauchen ca. 75 m^2 Wohnung mit 2 Zimmer, Bad, Kuche, toilette und garage.
Target: Ich brauche ca. 75 m^2 — eine Wohnung mit 2 Zimmern, Bad, Küche, Toilette und Garage.

Explanation:
The word 'Zimmer' is replaced by 'Zimmern' because in German, the dative case is needed for indicating the number of rooms in this context.
Error type: case

Mistakes due to imprecise knowledge of verb position in German

Source: in dem Sinn, möchte ich dazu sagen, dass integrationsprozes entwickelt sich schneller, wenn man das Gastland nicht als „fremd“ empfindet.
Target: In dem Sinn möchte ich dazu sagen, dass ein Integrationsprozess sich schneller entwickelt, wenn man das Gastland nicht als fremd" empfindet.

Explanation:
The word 'entwickelt' is relocated because the verb should be positioned after the subject in a German sentence.
Error type: word order

Mistakes due to other reasons

Source: Schauen Sie zuerst ihre Auswahl, sprechen Sie direkt mit dem Verkäufer.
Target: Schauen Sie zuerst ihre Auswahl an, sprechen Sie direkt mit dem Verkäufer.

Atomic edit:
["insert", "", "an"]

Explanation:
The word 'an' is inserted because the verb 'schauen' requires the preposition 'an' when it means 'to look at' something.
Error type: preposition

Table 10: Examples of wrong error explanations due to different reasons. For each sentence pair, only the explanations with a wrong error explanation are listed. All grammar errors in the target sentences are from the corpora.

Works on feedback comment generation. Among the work in feedback generation, Hanawa et al. (2021) is probably the most similar one to our GEE task because it aims to provide feedback comments to both preposition and general errors. However, its setting is the same as Nagata et al. (2021) and aims to offer feedback comments for a language user to correct their own errors. As a result, the feedback may not directly point out the correct usage of a word or a correct collocation. Other works in FCG solely focus on preposition words. Coyne (2023) and Coyne et al. (2023) work on the FCG task and develop a typology for learning feedback, including abstract types (e.g., tone and idiom) and grammatical pattern types (e.g., comparative and causative). However, their work is in an early stage with no human or automatic evaluation on the comment quality. Behzad et al. (2023) present a strong baseline for the FCG task but points out that, at the current stage, many feedback comments are generic (e.g., *Look up the use of the <verb> X in a dictionary and rewrite the sentence using the appropriate structure.*) Stahl and Wachsmuth (2023), Jimichi et al. (2023), and Ueda and Komachi (2023) approach the FCG task via fine-tuning language models such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020). However, for the GEE task, especially when there are restricted annotated resources for fine-tuning, it is unclear whether such an approach can work. Lastly, these works evaluate model outpus with BLEU (Papineni et al., 2002) and lack careful human evaluation.

K Error types generated by GPT-4

Table 11 and Table 12 list the frequent error types generated by GPT-4 in the German and Chinese GEE task.

Error Type	Count	Percent	Error Type	Count	Percent
punctuation	520	16.48	abbreviation	8	0.25
spelling	470	14.89	compound noun	8	0.25
capitalization	353	11.19	noun form	7	0.22
gender and case	175	5.54	extra word	6	0.19
preposition	163	5.16	syntax	6	0.19
word order	157	4.97	adjective	6	0.19
case	119	3.77	adverb	6	0.19
determiner	100	3.17	word form	6	0.19
adjective inflection	71	2.25	verb tense	6	0.19
verb conjugation	62	1.96	noun	5	0.16
conjunction	59	1.87	spelling and capitalization	5	0.16
pronoun	39	1.24	tense	5	0.16
conjugation	33	1.05	comparative	5	0.16
verb form	30	0.95	formatting	5	0.16
word choice	30	0.95	word formation	5	0.16
redundancy	30	0.95	possessive pronoun	4	0.13
plural	29	0.92	preposition and case	4	0.13
infinitive	29	0.92	time expression	4	0.13
unnecessary word	26	0.82	possessive	4	0.13
vocabulary	26	0.82	auxiliary verb	4	0.13
subject-verb agreement	25	0.79	demonstrative pronoun	4	0.13
article	22	0.70	idiomatic expression	4	0.13
verb	20	0.63	missing subject	4	0.13
adjective agreement	20	0.63	past participle	4	0.13
reflexive pronoun	19	0.60	spacing	4	0.13
gender	16	0.51	separable verb	4	0.13
expression	13	0.41	negation	4	0.13
subject	13	0.41	modal verb	4	0.13
compound word	12	0.38	terminology	4	0.13
missing word	11	0.35	relative pronoun	4	0.13
adjective form	11	0.35	singular/plural	4	0.13
plural form	11	0.35	gender agreement	4	0.13
subject omission	10	0.32	compound verb	4	0.13
verb choice	10	0.32	verb agreement	4	0.13
missing verb	8	0.25	spelling and inflection	4	0.13
translation	8	0.25	compound separation	4	0.13

Table 11: A distribution over error types in German grammatical error explanations (3156 total points, types with 4 or more datapoints considered). Overall, we observe a wide variety of error types.

Error Type	Count	Percent	Error Type	Count	Percent
word choice	588	39.65	extraneous word	7	0.47
redundancy	120	8.09	unnecessary ‘的’	7	0.47
word order	101	6.81	preposition usage	7	0.47
missing word	55	3.71	subject omission	6	0.40
miswritten character/word	52	3.51	‘们’	5	0.34
usage of ‘了’	44	2.97	missing particle	5	0.34
"de" particles	31	2.09	redundant character	5	0.34
preposition	24	1.62	redundant ‘的’	5	0.34
redundant word	22	1.48	emphasis	5	0.34
conjunction	21	1.42	particle usage	4	0.27
omission	20	1.35	redundant phrase	4	0.27
verb-object collocation	19	1.28	auxiliary verb	4	0.27
word omission	18	1.21	modal verb	4	0.27
unnecessary word	17	1.15	missing verb	4	0.27
sentence structure	15	1.01	unnecessary particle	4	0.27
usage of ‘的’	14	0.94	conjunction/connective	3	0.20
extra word	11	0.74	missing words	3	0.20
grammar	9	0.61	idiomatic expression	3	0.20
missing information	9	0.61	aspect particle	3	0.20
conjunction usage	8	0.54	unnecessary character	3	0.20
missing subject	8	0.54	adverb usage	3	0.20
measure word	8	0.54	expression	3	0.20
negation	8	0.54	unnecessary use of ‘的’	3	0.20

Table 12: A distribution over error types in Chinese grammatical error explanations (1483 total points, types with 3 or more datapoints considered). Overall, we observe a wide variety of error types.