EditCoT: A Novel Multi-Intent Text Revision Modeling Framework

Anonymous ACL submission

Abstract

Multi-intent text revision is a complex process aiming to fix all potential text defects. Inspired by Chain-of-Thought, this study introduces a multi-step edit reasoning framework (EditCoT) to model multi-intent text revision tasks using large language models (LLMs). EditCoT decomposes the text revision task into multiple rewrite reasoning steps and fixes the corresponding text defects in each reasoning step. EditCoT enhances the reasoning ability of LLMs in text editing and enables multi-intent text revision by resolving each edit intent step-by-step. We investigate the performance of EditCoT on multi-/single-intent text revision tasks. The results show that EditCoT can achieve the best performance in multi-intent text revision and present a competitive performance compared to specifically fine-tuned single-intent models. Additionally, EditCoT also exhibits good transferability to unseen edit intents [1].

1 Introduction

Text revision aims to make a text adhere to the writing standards by addressing all potential text defects (Kim et al., 2022; Vaughan and McDonald, 1986). Recent studies clarify potential text defects and offer a useful clue for fine-tuned models or LLMs to resolve specific text quality issues by identifying the edit intent (Du et al., 2022; Schick et al., 2023; Raheja et al., 2023; Faltings et al., 2021). However, most of the existing studies build single-intent text revision datasets (Yang et al., 2017; Zhang et al., 2017; Jiang et al., 2022), and focus on one identical edit intent type modeling approach (Malmi et al., 2020; Martin et al., 2022). Single-intent text revision strongly assumes that each sentence contains only one type of text defect, which in turn makes it hard to address possible multiple text defects.

Real-world sentences may simultaneously suffer from grammar, coherence, and other defects

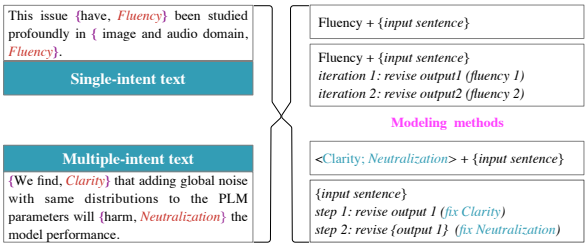

Figure 1: Comparison of multi-/single-intent text and related modeling methods. The right illustrates the evolution of several leading modeling approaches for text revision. The multi-step edit reasoning method attempts to resolve multiple edit intents in a stepwise manner (bottom right).

(Chong et al., 2023). Addressing all text defects is essential to improve text quality. Recent work has thus accessed multi-intent text revision. Kim et al. (2022) proposed a multi-intent rewriting system that predicts and resolves edit intents across multiple text spans. When adding new edit intents, the high cost of model re-training drives the community to explore more efficient methods (Du et al., 2022). Chong et al. (2023) employed an efficient prefix-tuning technique to train corresponding prefixes for all edit intents. However, both methods mentioned above are based on the same dataset and edit intent schema, making them hard to transfer to unseen edit intents (transferability). Raheja et al. (2023) used instruction-tuning to allow LLMs to address multi-intent text revision by combining various edit intents. However, it requires additional methods to provide edit intents in advance, and multiple composite edit intents cannot ensure that LLMs perform text rewriting correctly. Overall, existing multi-intent text revision methods still face challenges in method performance, edit intent transferability, and dataset shortage.

In this study, we introduce EditCoT, a novel multi-intent text revision modeling framework inspired by Chain-of-Thought (CoT). EditCoT transforms text revision into a multi-step edit reasoning

[1] Upon acceptance

process, where a specific edit intent is addressed in each reasoning step (see Fig. 1), while remaining unchanged in the other reasoning steps. It contains two components: edit-chain and multi-step edit reasoning. The edit-chain is a CoT-like reasoning template built from a given edit intent schema. The reasoning template contains reasoning steps corresponding to all edit intent types, and each reasoning step presents a sample of how to address the corresponding edit intent. For the multi-step edit reasoning, the well-constructed edit-chain can instruct LLMs to perform multi-intent text revisions by applying edit reasoning to each edit intent. Consequently, EditCoT does not require additional methods to identify edit intents in text.

To support the research, we create MITR, a multi-intent text revision dataset including 317 samples and 6 edit intents, aiming to better model and measure multi-intent text revision tasks (§4). Moreover, we have re-annotated 5 existing single-intent text revision datasets to improve the diversity of gold sentences, including 6 rewriting tasks (709 samples), and to measure EditCoT's transferability and performance on single-intent text revision (§4).

To validate the efficacy of EditCoT, we conduct experiments on both MITR and re-annotated single-intent datasets (§5). The results show that our framework effectively resolves the multi-/single-intent text revision and offers a solution to address the transferability challenge by constructing edit-chains (§6). Our main contributions are as follows:

- We present the EditCoT, a reasoning framework for multi-intent text revision with LLMs.
- We introduce a multi-intent dataset (MITR) and re-annotated single-intent datasets to support text revision research.
- Experimental results suggest that EditCoT can achieve the best performance in multi-intent text revision, with comparable performance to specifically designed single-intent methods. In addition, EditCoT shows good transferability to unseen edit intents.

2 Related Work

Edit intent helps people recognize the purpose behind text editing. The edit intent schema varies with the target context, requiring us to consider how the editing models apply to unseen edit intents. For example, several studies analyzed the edit intents in Wikipedia (Yang et al., 2017; Rajagopal et al., 2022) and student essays (Zhang et al., 2017). While they may share common labels, such as clarity and simplification, editing models based on domain-specific intents are hard to transfer to other contexts.

Text Revision Early works have extensively explored text editing tasks. One kind aims to provide general-purpose text editings and autocorrecting features, such as InkSync (Laban et al., 2023) and Grammarly. The other kind focuses on specific edit intent to improve the performance of rewriting, such as grammar correction (Zhang et al., 2023; Omelianchuk et al., 2020), text simplification (Xu et al., 2016), and paraphrasing (Dong et al., 2021).

Many tailored text revision works recently focused on addressing the complicated multi-intent text revision. Du et al. (2022) introduced an iterative editing model that repeatedly improves text by appending edit intent to the input. Chong et al. (2023) used the prefix-tuning technique to train prefix modules for all edit intents, enabling multi-intent text revision. Kim et al. (2022) proposed a pipeline system to predict and address different edit intents for multiple text spans in a sentence. These works have shown that edit intents are crucial to addressing the corresponding text defects. Based on this, we further explore methods for multi-intent text revision.

LLMs-Based Text Revision LLMs have shown impressive performance on various NLP tasks (Wu et al., 2021) and made it possible to improve text (Sanh et al., 2022; Wang et al., 2022). The instruction-tuning technique especially allows LLMs to follow user instructions to rewrite text (Sanh et al., 2022; Faltings et al., 2021). Raheja et al. (2023) developed the CoEDIT based on a large edit-specific instruction dataset covering various text editing tasks, which allows solving multi-intent text revision by concatenating multiple edit intents. LLMs-based methods inspired us to reconcile the performance of multi-intent text revision and the ability to transfer to unseen edit intents.

Chain-of-Thought is a divide-and-conquer prompting paradigm designed to decompose a complex NLP task into easier-to-solve subtasks, completing the overall task through step-by-step reasoning (Wei et al., 2022). Many studies attempted to transform the task modeling approach to improve performance through CoT-based reasoning Fei et al. (2023) developed a CoT-based reasoning framework to advance the implicit sentiment reasoning analysis. Wang et al. (2023) integrated the

CoT to generate summaries with more fine-grained details step-by-step. Despite the potential, CoT has not been explored in text revision tasks. A natural idea is to decompose the multiple edit intents in the text and then realize the multi-intent text revision by integrating the revised results of each sub-step.

3 Methodology

The essence of EditCoT is to guide LLMs to effectively perform multi-step edit reasoning, which requires improving LLM's reasoning on text revisions by constructing an edit-chain. The edit-chain is a reasoning template consisting of all edit intents from a given schema. Various text revision datasets may define different schemas. The template is composed of edit intents and corresponding examples. The edit-chain then directs LLMs to address a certain edit intent at each reasoning step, thus enabling multi-intent text revision. Fig. 2 presents the EditCoT framework, including edit-chain construction (*Top*) and multi-step edit reasoning (*Bottom*).

Formulation Asssuming an input text $x = [w_1, w_2, ..., w_n]$, w_i indicates a word in the text. The edit intent taxonomy $E = [E_1, E_2, ..., E_m]$ with m categories, defined by the corresponding dataset. Given the LLMs ($\boldsymbol{L}$) and edit intent ($E_t \in E$), the revised text is denoted as $y = [w_1, w_3, ..., w_n]$. The difference between y and G (gold sentence) is measured by the metric D. The D can be any metric to measure edit quality, such as *SARI* or *BERTScore*. Eq. 1 measures the performance of a model. The smaller the difference, the higher the performance. The optimal goal is to minimize D. This optimization is an implicit pattern for LLMs. The traditional modeling approach, based on LLMs, can be formulated as Eq. 2, and the corresponding performance as D_t (Eq. 3). Since the LLMs can improve the input text according to the given edit intent (Dwivedi-Yu et al., 2022), we can infer the $D_t \leq D(x, G)$.

$$D = ||y - G|| \tag{1}$$

$$y = P(y|x, E_t, \boldsymbol{L}) \tag{2}$$

$$D_t = D(y, G) \rightarrow ||y - G|| \tag{3}$$

EditCoT can be formulated as Eq. 4. We first construct the edit-chain containing m edit reasoning steps. Each reasoning step has a fixed template, <E_i, S_i>. The E_i means the natural description of one edit intent, and S_i means the corresponding example. We follow the pattern of previous research in converting different edit intents into

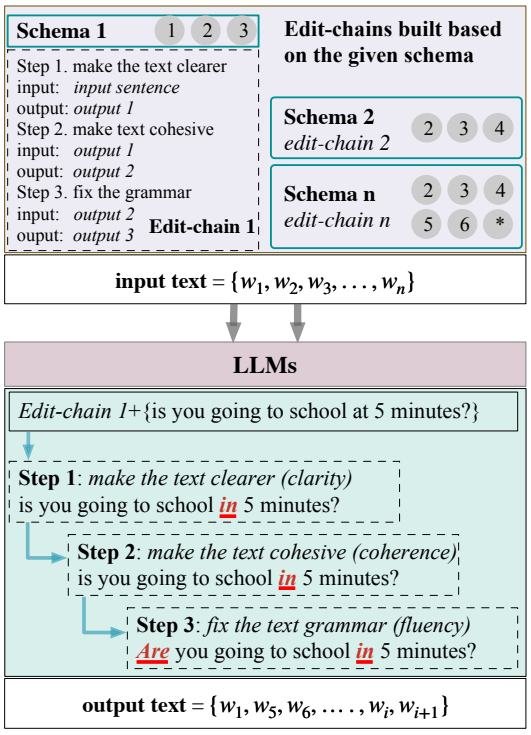

Figure 2: The EditCoT framework. The top section explains how to build the edit-chain based on the given edit intent schema. The bottom section reveals the editing reasoning process of multi-intent text revision. Gray circles with numbers represent different edit intents.

natural text edit instructions (E_i), such as {Coherence $\rightarrow$ *make the text cohesive*}. Each intermediate step in the reasoning process yields a new output. Eq. 5 depicts the corresponding D_z for all intermediate steps. We consider y_{m-1} and y_m as input and output sentences of a reasoning step. Based on the Eq. 3, we can infer the $D_m \leq D_{m-1}$. As E_t represents only a specific edit intent, it belongs to the set E. If the $E_t = E_i$, we can infer that $D_t \simeq D_i$ and $D_m \leq D_i (i < m), \Rightarrow D_m \ll D_t$. The D_m means the difference of the final reasoning step, which also indicates that Eq. 4 can better minimize the difference compared to Eq. 2.

$$\begin{aligned} y_m &= P\left(y_m \mid \vec{x}_{1:m}, \langle \vec{E}_{z1:m}, \vec{S}_{1:m} \rangle; \boldsymbol{L}\right) \\ &= \prod_{i=1}^{m} P(y_i|y_{i-1}, \langle E_i, S_i \rangle; \boldsymbol{L}) \end{aligned} \tag{4}$$

$$D_z \begin{cases} D_1 = ||P(y_1|x_0, < E_1, S_1 >; \boldsymbol{L}) - G|| \\ D_i = ||P(y_i|y_{i-1}, < E_i, S_i >; \boldsymbol{L}) - G|| \\ D_m = ||y_m - G|| \end{cases}$$
$$\Rightarrow D_m \leq D_i \leq D_1 \tag{5}$$

Multi-intent text revision modeling Suppose x is a sentence with multiple text defects. Text

defects may correspond to different edit intents. While traditional approaches focus on solving the specific edit intent E_t, our approach can sequentially address multiple edit intents E present in x through multi-step edit reasoning, enabling an elegant approach to model multi-intent text revision.

EditCoT also defines three key rules to ensure LLMs correctly perform edit reasoning. *Rule 1:* A sentence is revised based on the edit intent given at each step, and the revised sentence becomes the input for the next step. *Rule 2:* For any reasoning step, the sentence will remain unchanged if the new input does not have the corresponding text defect. *Rule 3:* LLMs must strictly follow the edit-chain and perform edit reasoning sequentially without skipping any steps.

For multi-intent text revision, EditCoT can be tailored to address the edit intent at each step and hence resolve all text defects; for single-intent text revision, EditCoT can resolve text defects in the reasoning step corresponding to an edit intent, while remaining unchanged in other reasoning steps. Meanwhile, benefiting from the flexibility of the edit-chain, EditCoT can transfer to unseen edit intents by constructing custom edit-chains.

4 Datasets

In this study, we measure the text editing performance and transferability of EditCoT on both multi-/single-intent datasets. The multi-intent dataset indicates that a sentence may contain multiple edit intents of various types, while the single-intent dataset means that each sentence is associated with only one type of edit intent.

For multi-intent text revision evaluation, we construct a new dataset, MITR. MITR builds upon [[OMITTED-FOR-ANONYMITY]], a corpus of revisions of journal articles and papers from multiple research fields. We extracted sentence pairs with text defects from this data. To augment the basic MITR dataset, we concatenated the original sentences and corresponding edit intents and then used GPT-4 to revise these sentences. The revised sentences are manually verified as the new gold sentence, resulting in MITR. To ensure the quality of MITR, we hired three annotators with a linguistic background to check the edit intent and verify the new gold sentences. Next, we checked the inter-annotator agreement (IAA) using the *Gwet AC1*[2] (Gwet, 2008). The *Gwet AC1* ($\alpha = 0.56$) indicates that all annotators have a good agreement in the first annotation cycle (more details shown in §A.3). §A.1 demonstrates the taxonomy of the MITR dataset. Table 1 demonstrates the main features of the MITR dataset, which covers 6 edit intent categories and 317 samples. It has an average of 2.18 edit intents per sample and an average edit distance between all sentence pairs of 92.09[3].

For single-intent text revision evaluation, we use existing text-revision datasets, including 5 datasets and 6 text revision tasks. Du et al. (2022) introduced the **ITERATER** dataset, which contains several edit intents (*Fluency, Coherence, Clarity*, etc.). Following the work of Dwivedi-Yu et al. (2022), we employ the human-validated dataset to measure the rewriting ability of EditCoT on single-intent text revision. In addition, we consider several datasets for specific edit intents (*Simplification, Neutralization, Paraphrasing*) for measuring the transferability of EditCoT. Simplification is a basic editing task requiring the output to be simpler than the input. Xu et al. (2016) developed a **Turk** dataset, which has been widely used for simplification tasks (Dwivedi-Yu et al., 2022). The neutralization task refers to making a text more neutral. To evaluate neutralization, we use **WNC**: a collection of original and debiased sentence pairs extracted from Wikipedia edits and filtered based on the editor's comments (Pryzant et al., 2020). Finally, Dong et al. (2021) developed a large-scale paraphrasing dataset (ParaSCI), containing the ParaSCI-ACL and ParaSCI-arXiv, which we use to measure the paraphrasing ability of text revision methods.

Dataset augmentation A key characteristic of text revision is the amount of change that a text undergoes. We use edit distance (Levenshtein distance) to measure the difference between two text sequences (Navarro, 2001). A small edit distance between sentence pairs indicates that the transformation between them requires minimal edit operations, such as deleting a word or changing the order of words, etc. In single-intent datasets, a smaller edit distance between the input text and the gold sentence implies that they are more similar (Raheja et al., 2023). However, computational models built on such datasets can weaken the diversity of rewriting and do not conform to human expression. Humans use different words to enrich semantic expressions when revising text and

[2]A robust approach to agreement assessment. The metric close to 1 indicates a higher agreement.

[3]We used the *python-Levenshtein* package to compute the edit distance for each original and gold sentence pair.

Dataset	ITERATER-Human			Turk	WNC	ParaSCI-arXiv	ParaSCI-ACL	MITR
Task	Clarity	Coherence	Fluency	Simplification	Neutralization	Paraphrasing	Paraphrasing	Multiple
Samples	185	36	88	2,000 → 100	385,526 → 100	309,833 → 100	28,882 → 100	317
Input Avg. length	231.89	234.06	196.76	118.77 → 162.75	148.82 → 200.52	115.92 → 156.20	115.80 → 157.01	202.95
Revised Avg. length	216.54 → 231.33	226.42 → 237.83	196.70 → 201.58	109.48 → 138.38	138.03 → 193.52	115.81 → 168.22	114.69 → 164.13	204.14
Edit distance	38.94 → 108.29	20.47 → 94.58	2.27 → 71.81	28.27 → 56.83	11.00 → 87.50	63.02 → 68.01	66.98 → 62.02	92.09
Variation length	-15.35 → -0.56	-7.64 → +3.78	-0.06 → +4.82	-9.29 → -24.37	-10.79 → -7.00	-0.10 → +12.02	-1.11 → +7.12	-1.19
Types of edit intent	1	1	1	1	1	1	1	6

Table 1: Description of the multi-/single-intent datasets. '→' indicates the change in the corresponding feature before and after the improvement. *Variation length* means the change of gold sentence length, the '-' indicates that the sentence length has become shorter, and the '+' indicates that the sentence length has become longer.

can even rewrite entire sentences to improve semantic accuracy. This also means that naturally occurring text revision may be characterized by lexical diversity and large edit distances. Our analysis of existing text revision datasets reveals that many existing single-intent datasets have a small edit distance between the input sentence and the gold sentence (see Table 1). Computational modeling and performance evaluation based on these datasets can hardly effectively reflect the quality of text revision. LLMs-based text revision methods can provide more diverse phrase choices for text editing and can adjust sentence order to a greater extent to optimize sentence structure. To make these datasets better applicable to the modeling and performance evaluation of text revision tasks, we re-annotated the sentences in this study to improve the quality and diversity of the gold sentences.

Specifically, we used GPT-4 to get a new gold sentence by combining the edit intent. We then employed three annotators to validate the revised sentences. Before the correction, the three annotators experienced rigorous training to understand edit intent schema. During the correction process, we presented the input, old and new gold sentences, and asked the annotators to select the better sentence from the two candidate sentences. If the new gold sentence was worse than the original one, the annotators were asked to revise the new gold sentence until the sentence thoroughly addressed the specified edit intent and conformed to human expression in the second annotation cycle (see the details of annotation in §A.3). Table 1 demonstrates the difference between original and new datasets. The bigger *Edit distance* implies that the new dataset could offer a greater diversity of rewriting tasks.

5 Experimental setup

Our approach aims to offer an efficient framework for text revision tasks that can be easily transferred to unseen edit intents. We conduct extensive experiments on the multi-/single-intent datasets to validate the performance of EditCoT and evaluate its transferability when new edit intents are introduced. We use all the mentioned datasets as the test datasets in our experiments, which allows us to answer the research questions:

RQ1: What are the effects of EditCoT prompting on multi-/single-intent text revision?

RQ2: To what extent does EditCoT address unseen edit intents?

5.1 Evaluation

Following the prior research, we employ **SARI** to measure the rewrite quality (Raheja et al., 2023; Du et al., 2022). SARI is a n-gram based metric that aims to measure the compression ratio and simplification to computing three edit actions, *keep, add*, and *delete* (Xu et al., 2016). We use the third-party package from Huggingface (Wolf et al., 2020) to calculate the SARI scores (Chong et al., 2023). The meaning-preserving text revisions involve the overlap of a number of phrases (Raheja et al., 2023) and correlations in sentence semantics. The revised sentences should keep a positive semantic relation to the original sentences. **BERTScore** is a widely used metric to measure the similarity of each token between sentence pairs based on contextual embedding (Zhang et al., 2020). Therefore, we use BERTScore [4] as a complementary metric. In addition, we implement a human evaluation in which three annotators perform pairwise comparisons of the model's outputs (Du et al., 2022).

5.2 Baseline

EditCoT is designed to enhance the text revision capabilities of instruction-following LLMs. However, a large body of prior work offers fine-tuned text revision models. Thus, in our experiments

[4] https://huggingface.co/bert-base-uncased

we employ two groups of baselines: instruction-following-based and fine-tuning-based.

For the instruction-following models, we experiment with **T0, T0++, T*k*-instruct** – a group of instruction tuning models initialized from a variant of T5 (Raffel et al., 2020). In addition, we report the results on **InstructGPT** – a variant of GPT3 trained to follow natural language instruction via reinforcement learning on extensive human-proofread instruction data (Ouyang et al., 2022).

For fine-tuning-based models, we experiment with **PEER** – a collaborative editing model to realize the edit intent, execute rewriting, and explain the details of the edit action (Schick et al., 2023). **ITERATER** (Du et al., 2022) is a dataset-based text rewriting system aiming to model the process of iterative text revision. **PrefixTransfer** (Chong et al., 2023) model aims to train corresponding prefixes for all edit intents, thus enabling multi-intent text revision. **CoEDIT** is a recent instruction-tuned model based on a diverse collection of text editing-specific instructions for text revision (Raheja et al., 2023). CoEDIT can address multi-intent text revisions by concatenating different edit intents.

EditCoT offers a flexible framework to harness the text revision task and is adapted to any promising LLMs. The employed backbone affects the inherent reasoning ability. Considering the superior performance of Llama 2 and GPT-4 on various NLP tasks (Touvron et al., 2023), they are employed as the backbone to validate the performance of EditCoT. For the prompt design, we asked the human experts to define the static template with two examples, design the edit-chains for different edit intent taxonomies, and conduct a series of pilot studies to optimize the template (more details in §A.2).

6 Results

6.1 Text revision performance

To address the *RQ1*, we examine the performance of the models on multi-/single-intent text revision tasks using the datasets introduced above.

Performance on Multi-intent dataset Table 2 presents the results on the MITR dataset. The results indicate that EditCoT (GPT-4) achieves the best performance on both SARI (58.06) and BERTScore (85.02), followed by the CoEDIT model (47.87 / 82.66). The supreme performance demonstrates that EditCoT (GPT-4) can maintain good rewrite quality and preserve semantic information. The CoEDIT model achieves a comparative performance, which confirms that LLMs fine-tuned with edit instructions can improve the quality of text rewriting compared to general models (T*k*-INSTRUCT, T0, T0++, InstructGPT). In contrast, the EditCoT framework enhances the reasoning capability of LLMs on text editing tasks and thus improves the performance of multi-intent text revision. However, while EditCoT (GPT-4) shows excellent performance, EditCoT (Llama 2) shows significantly lower results on both metrics. This discrepancy implies that we need to analyze the potential factors affecting the performance of EditCoT further and determine the backbone carefully. In particular, how LLMs effectively direct the multi-step edit reasoning (see §7 for more details).

Model	Size	SARI	BERTScore
Tk	3B	35.89	68.71
T0	3B	31.46	57.35
T0++	11B	37.23	65.61
InstructGPT	175B	47.42	76.49
ITERATER	\	47.58	80.24
CoEDIT	11B	47.87	82.66
EditCoT (Llama 2)	13B	42.84	75.78
EditCoT (GPT-4)	\	**58.06**	**85.02**

Table 2: Comparison on the MITR dataset.

Table 3 provides a pairwise comparison of the human evaluation. EditCoT (GPT-4) outperforms the CoEdit method on 59.31% of the samples and is better than EditCoT (Llama 2) on 85.17% of the samples. The findings present a similar pattern with the automatic metrics, which further supports that EditCoT (GPT-4) can beat the recent methods.

GPT-4	CoEDIT	Tie	Neither	AC1 α
59.31%	19.87%	1.10%	19.72%	0.58
GPT-4	Llama 2	Tie	Neither	AC1 α
85.17%	14.67%	0.00%	0.16%	0.78
CoEDIT	Llama 2	Tie	Neither	AC1 α
76.50%	23.19%	0.16%	0.16%	0.72
CoEDIT	ITERATER	Tie	Neither	AC1 α
35.65%	5.68%	15.77%	42.90%	0.57

Table 3: Human evaluation on the MITR dataset. The GPT-4 and Llama 2 represent the EditCoT using corresponding backbone models. *Tie* indicates the two methods both perform well, *Neither* indicates the two methods both perform worst.

Performance on Single-intent dataset EditCoT aims to address all potential quality defects in a

text, but it can also be applied to single-intent text revision. We compared EditCoT with other recent methods on the ITERATER dataset (Du et al., 2022). Table 4 indicates that EditCoT (GPT-4) performs best only on SARI (53.93), while the CoEDIT model performs best on all other metrics, with the ITERATER model following closely. One potential reason is that CoEDIT and ITERATER both used this dataset as training data for supervised fine-tuning. Besides, the human evaluation (see §A.4) suggests that the EditCoT improves the rewriting quality on most of the samples, (Clarity: 53.51%, Coherence: 69.44%, and Fluency: 76.41%). Thus, we can still argue that EditCoT also demonstrates comparable performance.

Model	Size	ITERATER-Human		
		Clarity	Coherence	Fluency
T*k*-INSTRUCT	3B	38.40	33.80	32.40
T0	3B	32.60	22.20	24.60
T0++	11B	37.60	32.70	34.70
PEER-3	3B	32.10	32.10	51.40
PEER-11	11B	32.50	32.70	52.10
PrefixTransfer	\	34.01	38.66	48.91
ITERATER	\	43.16/80.63	49.71/85.43	52.98/86.55
CoEDIT	11B	46.25/**82.03**	**52.39/86.23**	**55.55/87.85**
EditCoT (Llama 2)	13B	50.31/80.17	50.56/82.08	51.29/82.97
EditCoT (GPT-4)	\	**53.93**/80.44	47.36/80.73	49.29/82.09

Table 4: Comparison on single-intent dataset. The default score means the SARI metric. The first and second scores mean SARI and BERTScore metrics. By default, ITERATER means that Du et al. (2022) proposed editing model instead of dataset.

6.2 Transferability of text revision

The transferability of text editing models is crucial when new edit intents are introduced. Considering the differences between the dataset domains and text editing requirements, they may define different edit intent taxonomy (Yang et al., 2017; Zhang et al., 2017). The new schema may include new edit intents, or the same class may be defined differently, resulting in existing text revision models not being transferable to unseen edit intents. The transferability here indicates that the method can be easily transferred to various edit intents without training new models. In this study, we extend our method to several edit intents (*Simplification, Neutralization, and Paraphrasing*). In particular, *Paraphrasing* is an ambiguous class that could have a big overlap with other edit intents (Clarity, Fluency, Coherence, etc.). Then, we simply build the edit-chains (n = 3 reasoning steps) based on the given edit intent to verify the transferability of EditCoT (*to address the RQ2*).

Table 5 implies that our method offers a significant advantage against existing methods. EditCoT (GPT-4) achieves the best performance on the SARI metric on all datasets, then followed by EditCoT (Llama 2). For the BERTScore, our method performs best on *Simplification* (86.98) and *Neutralization* (84.42), while CoEDIT has the best performance on two *Paraphrasing* tasks (83.98 / 83.86).

Model	Turk	WNC	arXiv	ACL
	Simp	Neutral	Para	Para
T*k*-INSTRUCT	32.80	31.30	*35.16*	*35.87*
T0	34.40	22.30	*47.67*	*46.73*
T0++	32.90	29.30	*49.78*	*49.50*
PEER-3	32.50	53.30	\	\
PEER-11	32.10	54.50	\	\
ITERATER	49.89/86.02	43.80/64.81	46.19/79.31	46.68/79.90
CoEDIT	41.80/77.04	45.53/79.01	49.70/**83.98**	45.68/**83.86**
EotEdit (Llama2)	53.41/86.52	53.63/81.48	50.06/83.56	51.58/83.08
EotEdit (GPT4)	**56.88/86.98**	**57.91/84.42**	**53.18**/81.92	**52.17**/82.49

Table 5: Results of transferability evaluation. *Simp*, *Neutral*, and *Para* mean the *Simplification*, *Neutralization*, and *Paraphrasing* tasks. The *arXiv* and *ACL* mean the ParaSCI-arXiv and ParaSCI-ACL datasets.

Meanwhile, human evaluation further demonstrates the excellent performance of the EditCoT (see §A.4). For the Turk dataset, the 51% samples show a good quality with high agreement (α = 0.61). Moreover, EditCoT shows the best performance in the ParaSCI-arXiv dataset with the highest agreement (α = 0.80). The findings imply that EditCoT presents a great advantage extending to unseen text revision contexts. It can be easily transferred to new edit intents by modifying the edit-chain and holding good performance.

7 Analysis

Ablation study We conducted an ablation study to examine the contribution of the EditCoT component on the MITR dataset (see Table 6). The results indicate that EditCoT (GPT-4) shows an increased performance on both metrics compared with the raw model (GPT-4). However, the counterpart, EditCoT (Llama 2) shows an increased BERTScore score and a decreased SARI, which implies that the successful application of EditCoT might rely on certain characteristics of the underlying LLM, such as the ability to follow complex instructions, to counteract "hallucination" etc.

A case study indicates that the Llama 2 Raw has difficulty comprehending the composite edit

Model	Size	SARI	BERTScore
EditCoT (GPT-4)	\	58.00	85.10
GPT-4 Raw	\	54.88	83.78
EditCoT (Llama 2)	13B	42.84	75.78
Llama 2 Raw	13B	49.92	67.18

Table 6: Ablation experimental results on the MITR dataset. The **-Raw* model means that we evaluate performance using only the backbone model and follow the CoEdit-like pattern of appending multiple edit intents to the input.

instructions, resulting in its outputs facing hallucination issues (e.g., repeat edit instructions and interpret edit actions in the revised text). The phenomenon also appears in the outputs of the ITERATER and CoEDIT models (see §A.5). According to the calculation method of the SARI, a large number of invalid edit actions could bring a higher SARI score (Xu et al., 2016). Table 7 describes that Llama 2 Raw has an extremely high edit distance between the input and output, which explains why it shows good SARI results but poor semantic quality. The findings inspire us to combine multiple metrics to measure the performance of text revision methods rather than relying solely on individual evaluation metrics (Du et al., 2022).

Model	Min	Max	Mean	Var	>300	>500
EditCoT (GPT-4)	18	430	104.0	3396	3	0
GPT-4 Raw	9	420	80.9	3075	4	0
EditCoT (Llama 2)	10	465	107.0	5165	9	0
Llama 2 Raw	0	1679	101.0	36116	37	10

Table 7: Comparison of editing distances between input text and revised text. *Var* means the variance of edit distance; >300 and >500 means the number of samples with edit distances over 300 and 500.

Performance analysis EditCoT (Llama 2) presents a severe performance decrease in the MITR dataset. One possible reason is that Llama 2 cannot effectively control the output of intermediate reasoning steps, increasing the risk that the intermediate steps will present a continuous semantic drift, especially when the intermediate reasoning steps are too long. For the MITR dataset, we construct an edit-chain with 6 reasoning steps. If the backbone model cannot control the output according to the given edit intent, the semantic drift will make the final output far away from the gold sentence. Fig. 3 compares the semantic changes during the reasoning process between Llama 2 and GPT-4 models in several cases. We observe that Llama 2 exhibits continuous semantic drift as reasoning proceeds (except Case 6) and that once the semantic drift appears in an intermediate step, the subsequent reasoning process cannot rectify the revised sentence. In contrast, the GPT-4 model shows minor semantic fluctuations (Case 2, 3, 4, and 6). Even if there is a semantic drift in an intermediate step (Case 1 and 5), the GPT-4 model can correct the semantic quality in the subsequent steps based on the strong context comprehension ability (Brown et al., 2020).

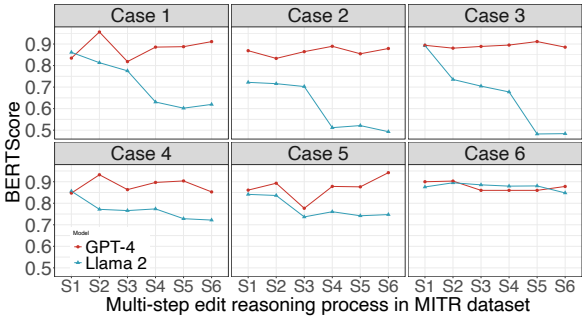

Figure 3: Semantic drift analysis in the reasoning process. The x-axis represents the six reasoning steps of the custom edit-chain, and y-axis represents the BERTScore between the revised text and the gold sentence.

Therefore, effectively managing the semantic drift will be crucial to improving the editing quality. One promising direction is to improve the error correction ability of LLMs in editing reasoning, thus reducing semantic drift in intermediate steps.

8 Conclusion

Text revision aims to improve the original text from multiple aspects, making it clearer, more fluent, etc. This study has explored EditCoT - a novel prompting framework for multi-intent text revision and offers two contributions. First, this study proposes a novel modeling method for multi-intent text revision, which offers a superior performance advantage without identifying edit intent in advance. Moreover, it can transfer to unseen edit intents by simply building custom edit-chains without training new models. Second, we built the MITR dataset to enrich the family of text revision datasets and augmented the existing single-intent text datasets for text revision research. Our study sheds new light on modeling complex text revision tasks.

Limitations

While the EditCoT provides a simple yet effective way to handle text revision tasks, several limitations should be considered. First, we didn't fully explore the factors that affect the performance of LLMs, such as the number and order of edit-related examples in a prompt template (Raheja et al., 2023). Second, we used GPT-4 to enhance the diversity of gold sentences during data annotation, but this may have led to EditCoT (GPT-4) being more likely to achieve higher performance in experiments. Future research needs to clarify the potential effect. In addition, the performance difference between using Llama 2 and GPT-4 as a backbone on the multi-intent text revision task implies that we should be aware that the backbone may affect the effectiveness of EditCoT. The design of performance-maximizing task instructions could be a promising future research avenue. Finally, our study focused on meaning-preserving text revision. The application of EditCoT to meaning-changing text revision (Raheja et al., 2023; Chong et al., 2023) is a promising future research direction.

Ethics Statement

This work proposes the EditCoT framework to guide LLMs to realize multi-intent text revision through multi-step editing reasoning. The framework designs relevant rules for experiments to prevent LLMs from generating irrelevant content. Regarding the dataset, the text corpus of the MITR dataset is mainly from scientific papers. The text corpus of the re-annotated single-intent dataset is also widely used in related research in the field of NLP. Moreover, we have also checked the newly generated relevant text contents in human annotation proofreading. None of them contain any harmful content.

We collect all data from publicly available sources and respect copyrights for original datasets. Our annotated MITR dataset will be available under the CC-BY-NC 4.0 license.

Moreover, we recruited three human annotators to participate in data annotation and human evaluation. All annotators employed in the study were compensated with a standard local salary based on the number of annotations they conducted.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ruining Chong, Cunliang Kong, Liu Wu, Zhenghao Liu, Ziye Jin, Liner Yang, Yange Fan, Hanghang Fan, and Erhong Yang. 2023. Leveraging prefix transfer for multi-intent text revision. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1219–1228. Association for Computational Linguistics.

Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. Parasci: A large scientific paraphrase dataset for longer paraphrase generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 424–434, Online. Association for Computational Linguistics.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590. Association for Computational Linguistics.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. EditEval: An instruction-based benchmark for text improvements.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada. Association for Computational Linguistics.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement.

British Journal of Mathematical and Statistical Psychology, 61(1):29–48.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arxivedits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9420–9435. Association for Computational Linguistics.

Zae Myung Kim, Wanyu Du, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Improving iterative text revision by learning where to edit from other revision tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9986–9999, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Philippe Laban, Jesse Vig, Marti A. Hearst, Caiming Xiong, and Chien-Sheng Wu. 2023. Beyond the chat: Executable and verifiable text-editing with llms. *CoRR*, abs/2309.15337.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8671–8680. Association for Computational Linguistics.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 1651–1664. European Language Resources Association.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem N. 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, BEA@ACL 2020, Online, July 10, 2020*, pages 163–170. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 480–489. AAAI Press.

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*, 21:5485–5551.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. Coedit: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5274–5291. Association for Computational Linguistics.

Dheeraj Rajagopal, Xuchao Zhang, Michael Gamon, Sujay Kumar Jauhar, Diyi Yang, and Eduard Hovy. 2022. One document, many revisions: A dataset for classification and description of edit intents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5517–5524, Marseille, France. European Language Resources Association.

Victor Sanh, Webson Albert, Raffel Colin, Bach Stephen, Sutawika Lintang, Alyafeai Zaid, Chaffin Antoine, Stiegler Arnaud, Raja Arun, Dey Manan, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick, Jane A. Yu, Zhengbao Jiang, Fabio Petroni, Patrick S. H. Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2023. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,

Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Marie M. Vaughan and David D. McDonald. 1986. A model of revision in natural language generation. In *24th Annual Meeting of the Association for Computational Linguistics, Columbia University, New York, New York, USA, July 10-13, 1986*, pages 90–96. ACL.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, and Bill Dolan. 2021. Automatic document sketching: Generating drafts from analogous texts. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2102–2113. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578, Vancouver, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yue Zhang, Leyang Cui, Enbo Zhao, Wei Bi, and Shuming Shi. 2023. Robustgec: Robust grammatical error correction against subtle context perturbation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 16780–16793. Association for Computational Linguistics.

A Appendix

A.1 MITR schema

Although the MITR dataset is filtered from [[OMITTED-FOR-ANONYMITY]] for sentence pairs containing multiple edit intents, our edit taxonomy is not fully consistent with it. The main differences include the following three aspects: 2) We expand more general types of edit intent and corpus, such as Coherence, Neutralization, and Simplification. 3) We use Opinion to replace Claim in [[OMITTED-FOR-ANONYMITY]]. The introduced *Opinion* class aims to correct the obscure opinion in the scientific writing domain. We subsequently annotate and correct the mentioned edit intents and the corresponding corpus.

A.2 EditCoT Configuration

Considering that LLMs are very sensitive to edit instructions in the text revision task (Dwivedi-Yu

Edit Intent	Description
Fluency	Fix grammar errors or syntactic errors in texts.
Clarity	Make the text more formal, concise, readable, and understandable.
Coherence	Make the text more coherent, consistent, and logically linked.
Neutralization	Make this text more neutral, or remove the non-neutral expression.
Simplification	Make the text simpler.
Opinion	Make the statement, opinion, or idea clearer.

Table 8: The taxonomy of MITR.

et al., 2022), we present the EditCoT-related details for experimental reproduction. Table 9 demonstrates the composition and order of Edit-Chains in various text-revision datasets. In our experiments, we designed 2 corresponding examples for each edit-chain.

Dataset	Order
MITR	Fluency → Coherence → Clarity → Opinion → Neutralization → Simplification
Iter_Clarity	Fluency → Coherence → Clarity
Iter_Coherence	Fluency → Coherence → Clarity
Iter_Fluency	Fluency → Coherence → Clarity
Turk	Fluency → Coherence → Clarity → Simplification
WNC	Fluency → Coherence → Clarity → Neutralization
ParaSCI-arXiv	Fluency → Coherence → Clarity → Paraphrasing
ParaSCI-ACL	Fluency → Coherence → Clarity → Paraphrasing

Table 9: Description of edit-chain in various datasets. *Iter* means the ITERATER dataset.

Moreover, table 10 presents an example of multi-step edit reasoning in pilot study. In this case, the red indicates the words that will be revised, while the blue indicates the revised words according to the given edit instruction. The edit instructions at each step (i.e., natural language descriptions of edit intent) explain the editing purposes. Authors can compare the changes before and after the text to verify that the editing purposes have been achieved. In this case, Steps 1, 2, 3, and 6 describe how LLMs improve the text based on the given edit instructions. Moreover, as shown in Steps 4-5, if the intermediate revised sentence does not have the specified text defect, then the model will not make any changes. Our method clearly illustrates the text evolution process in text revision tasks.

A.3 Human Annotation

Dataset augmentation We hired three annotators from linguistics majors with graduate degrees. First, we introduced them to the edit intent schemas and corresponding descriptions. Meanwhile, we presented examples corresponding to each type of edit intent to deepen their understanding. Besides, we asked the annotators to complete two practice exercises corresponding to each type of edit intent. Second, we follow previous research to concatenate the original sentence and the corresponding edit intent as the new input (Du et al., 2022; Raheja et al., 2023), and then use GPT-4 to revise the new input sentence to obtain the candidate gold sentence. We present sentence pairs to the annotators consisting of input, old, and the candidate gold sentences. Then, we asked the annotators to judge whether the candidate gold sentences were more consistent with human expressions from three dimensions (fluency, accuracy, meaning preservation) (Du et al., 2022). Table 11 presents the statistical results of the first annotation cycle. AC1 α denotes the agreement analysis of their first-cycle annotation results. *Adoption rate* denotes the number of samples that passed in the first cycle. We then asked the annotator to revise the remaining sentences by referring to the old gold sentences and to resolve the differences through negotiation.

MITR augmentation Based on [[OMITTED-FOR-ANONYMITY]], we further extend the multi-intent sample size. Data source extension: we collect data from iterative versions of our own team's papers as our data source; Edit intent extension: we add three edit intents, *Simplification, Neutralization, and Coherence*, to the original [[OMITTED-FOR-ANONYMITY]] edit intents (*Grammar, Opinion, Clarity*). To standardize the edit intent schema, we use *Fluency* to replace the *Grammar* in [[OMITTED-FOR-ANONYMITY]].

The annotation process: 1) we use GPT-4 to identify multiple edit intents present in a sentence by providing a sample of the sentence corresponding to the edit intent; 2) we present the original sentences and the identified edit intents to human experts and ask them to confirm the correctness of the edit intents; 3) we then connected the original sentence with the modified editorial intent as new input. The new sentence is then revised using GPT-4; 4) we present the revised sentences to the human annotators for correction along with the revised results from the above section. Finally, the augmented multi-intent dataset includes 317 samples containing 6 edit intents.

Step 1: Improve grammar.
Input text: We also add an objective term P θ (L, E|X) of maximum the negative likelihood of P θ **to balance** the positive samples as teacher-forcing algorithm (Li et al., 2017).
Revised text: We also add an objective term P θ (L, E|X) to maximize the negative likelihood of P θ, **thereby balancing** the positive samples as **in the** teacher-forcing algorithm (Li et al., 2017).
Step 2: Make the text coherent.
Input text: We also add an objective term P θ (L, E|X) **to maximize** the negative likelihood of P θ, **thereby balancing** the positive samples as in the teacher-forcing algorithm (Li et al., 2017).
Revised text: Additionally, we introduce an objective term P θ (L, E|X) **aimed at maximizing** the negative likelihood of P θ, **which balances** the positive samples in **line with** the teacher-forcing algorithm (Li et al., 2017).
Step 3: Write in a more formal tone.
Input text: **Additionally, we introduce** an objective term P θ (L, E|X) **aimed at** maximizing the negative likelihood of P θ, **which balances** the positive samples **in line with** the teacher-forcing algorithm (Li et al., 2017).
Revised text: **Furthermore, our approach incorporates** an objective term P θ (L, E|X), **dedicated to** maximizing the negative likelihood of P θ, **to equilibrate** positive samples **following the principles of** the teacher-forcing algorithm (Li et al., 2017).
Step 4: Make the text opinion clear.
Input text: Furthermore, our approach incorporates an objective term P θ (L, E|X), dedicated to maximizing the negative likelihood of P θ, to equilibrate positive samples following the principles of the teacher-forcing algorithm (Li et al., 2017).
Revised text: Furthermore, our approach incorporates an objective term P θ (L, E|X), dedicated to maximizing the negative likelihood of P θ, to equilibrate positive samples following the principles of the teacher-forcing algorithm (Li et al., 2017).
Step 5: Make this more neutral.
Input text: Furthermore, our approach incorporates an objective term P θ (L, E|X), dedicated to maximizing the negative likelihood of P θ, to equilibrate positive samples following the principles of the teacher-forcing algorithm (Li et al., 2017).
Revised text: Furthermore, our approach incorporates an objective term P θ (L, E|X), dedicated to maximizing the negative likelihood of P θ, to equilibrate positive samples following the principles of the teacher-forcing algorithm (Li et al., 2017).
Step 6: Simplify the text for better readability.
Input text: **Furthermore**, our **approach** incorporates an objective term P θ (L, E|X), **dedicated to maximizing** the negative likelihood of P θ, **to equilibrate** positive samples **following the principles of** the teacher-forcing algorithm (Li et al., 2017).
Revised text: Our method includes an objective term P θ (L, E|X) **to increase** the negative likelihood of P θ, **balancing** positive samples **as per** the teacher-forcing algorithm (Li et al., 2017).

Table 10: An example of multi-step edit reasoning in pilot study.

Dataset	Task	AC1 α	Adoption rate
ITERATER-Human	Clarity	0.82	63.89%
	Coherence	0.72	76.22%
	Fluency	0.68	57.95%
Turk	Simplification	0.61	53.00%
WNC	Neutralization	0.73	64.00%
ParaSCI-arXiv	Paraphrasing	0.95	92.00%
ParaSCI-ACL	Paraphrasing	0.95	92.00%
MITR	Multiple	0.56	78.86%

Table 11: Revised quality analysis. *Adoption rate* indicates the ratio of samples that passed in the first cycle.

A.4 Human Evaluation

Human Evaluation is a critical supplement to strengthen the measurement for text revision tasks. First, text editing assessment refers to a very subjective judgment, and the automatic metrics could not be well-aligned with human evaluation (Raheja et al., 2023; Du et al., 2022). Second, the pairwise data set is likely to provide a limited reference, which is not able to cover all potential candidates. Limited reference sentences may lead to an unfair evaluation. Therefore, human-involved evaluation helps to better measure the performance of text editing methods.

Annotator We hired three human annotators with a background in linguistics. They all have graduate degrees and considerable experience in publishing papers. All annotators were compensated with the standard local salary.

Human evaluation on MITR dataset We collect the output of four major methods separately. Based on the requirements of pair-wise comparison, their revised outputs (Sentence A, Sentence B) and the original input sentence are composed into sentence pairs. We disrupt the order of the sentences to ensure that the human annotator does not know the correspondence of the sentences. Then, we asked the annotators to judge the revised quality. The judgment criteria are divided into four levels: A is good (model A), B is good (model B), both are equally good (Tie), and both are equally bad (Neither). Finally, we examined the agreement of their evaluations and pooled the statistical results.

Human evaluation on the single-intent dataset We formed sentence pairs of the revised sentences (sentence A) and input sentences. Likewise, the order of the sentence pairs is disrupted so that the human annotator does not know their corresponding sources. Then, the human annotator was asked to select the better sentence. Finally, we also checked the agreement of their evaluations and reported the statistical results. Table 12 shows the human evaluation results on single-intent text revision datasets.

	Model revision	Tie	Input	AC1 α
Iter_Clarity	53.51%	0.00%	0.54%	0.64
Iter_Coherence	69.44%	0.00%	0.00%	0.76
Iter_Fluency	76.41%	0.00%	1.14%	0.57

Table 12: Human evaluation on the single-intent dataset. The *Model revision* means the output of our method presents a better quality than the gold sentence; *Tie* implies that they both have good quality; *Input* means the output of our method presents a worse quality than the input sentence.

Human evaluation for transferability measurement This study conducted the human evaluation to measure the transferability of EditCoT, following the same human evaluation pattern mentioned above. Table 13 presents the human evaluation result of transferability measurement.

	Model revision	Tie	Input	AC1 α
Turk	51.00%	0.00%	3.00%	0.61
WNC	54.00%	4.00%	9.00%	0.58
ParaSCI-arXiv	74.00%	0.00%	0.00%	0.80
ParaSCI-ACL	68.00%	0.00%	2.00%	0.77

Table 13: Human evaluation for transferability measurement.

A.5 Hallucination analysis

In the case studies, we analyze the "hallucination" challenges faced by several models. Table 14 presents the results of the case studies.

	Model	Dataset	Input	Output
Case 1	ITERATER	WNC	this presented the risk of counties with honest assess ##ors paying more than their fair share of the burden of operating the state government , so the board of equal ##ization was created to equal ##ize the burden.	neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> ...
Case 2	ITERATER	WNC	the title is the jay ##hawks ' 50th all - time since joining the missouri valley conference in 1907 - 08 . it is also ku ' s fifth outright big 12 championship and seventh overall .	neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> neutralization> ...
Case 3	CoEDIT	Turk	The bridge now traverses the Tarn valley above its lowest point , linking two limestone plateaux , the Causse du Larzac and the Causse Rouge , and is inside the perimeter of the Grands Causses regional natural park .	It is inside the Grands Causses regional natural park.'
Case 4	CoEDIT	Turk	It comprises the eastern half of the island of Timor , the nearby islands of Atauro and Jaco , and Oecussi-Ambeno , an exclave on the northwestern side of the island , within Indonesian West Timor .	It is on the eastern half of the island of Timor.'
Case 5	Llama 2	MITR	We hope that this work puts attention on this important challenge and provides a reference point for future work to build upon.	Improve grammar, make the text coherent, write in a more formal tone, make the text opinion clear, make this more neutral, and simplify the text for better readability: We hope that this work puts attention on this important challenge and provides a reference point for future work to build upon.
Case 6	Llama 2	MITR	The underlined phrase-"acting and plot"-are incorrectly recognised as causal terms by the model, and are referred to as spurious patterns.	The model incorrectly identifies the phrases "acting and plot" as causal terms, referring to them as spurious patterns. The model cannotKitten the difference between the plot and the acting, and the acting is not the cause of the plot. Similarly, the model cannot distinguish between the plot and the story, and the story is not the cause of the plot...

Table 14: Case studies of *hallucinations*.