Two-stage Incomplete Utterance Rewriting on Editing Operation

Zhiyu Cao, Peifeng Li*, Qiaoming Zhu, Yaxin Fan
School of Computer Science and Technology, Soochow University, Suzhou, China
zycao18@stu.suda.edu.cn, yxfansuda@stu.suda.edu.cn
{pfli, qmzhu}@suda.edu.cn

Abstract

Previous work on Incomplete Utterance Rewriting (IUR) has primarily focused on generating rewritten utterances based solely on dialogue context, ignoring the widespread phenomenon of coreference and ellipsis in dialogues. To address this issue, we propose a novel framework called TEO (*Two-stage approach on Editing Operation*) for IUR, in which the first stage generates editing operations and the second stage rewrites incomplete utterances utilizing the generated editing operations and the dialogue context. Furthermore, an adversarial perturbation strategy is proposed to mitigate cascading errors and exposure bias caused by the inconsistency between training and inference in the second stage. Experimental results on three IUR datasets show that our TEO outperforms the SOTA models significantly.

1 Introduction

Dialogue understanding (e.g., dialogue generation, dialogue sentiment analysis, and intent recognition) often suffers from ellipsis and coreference, because people often omit certain information or use pronoun in utterances for the sake of convenience in real-world conversations. Incomplete Utterance Rewriting (IUR) is to rewrite incomplete utterances more specific and direct, which is beneficial for many downstream dialogue understanding tasks, such as conversational dense retrieval (Qian and Dou, 2022) and dialogue summarization (Fang et al., 2022). Actually, IUR can be specifically categorized into coreference and ellipsis resolution. As shown in Table 1, the incomplete utterance u_3 uses the pronoun "*he*" to represent "*Ben Affleck*" and omits "*as Batman*". The rewritten utterance u_3' "*It is Ben Affleck who acted as Batman*" is more informative and direct than the original incomplete utterance "*It is he who acted*".

Speaker(turn)	Utterance
$\mathbf{Speaker_1(u_1)}$	I think Batman is very handsome.
$\mathbf{Speaker_2(u_2)}$	The poster looks a bit like Ben Affleck.
$\mathbf{Speaker_1(u_3)}$	It is **he** who acted. (Incomplete utterance)
$\mathbf{Speaker_1(u_3')}$	It is **Ben Affleck** who acted **as Batman**.

Table 1: An example of IUR, where **red** and **blue** color indicate coreference and ellipsis, respectively. The first two utterances u_1 and u_2 are dialogue history, the third u_3 is an incomplete utterance to be rewritten, and the fourth u_3' is the rewritten utterance.

Although previous methods have achieved great success, they only regarded IUR as a generation or sequence labeling task, which did not explicitly consider the two different operations of replacement (for coreference) and insertion (for ellipsis) (Hao et al., 2021; Chen, 2023; Li et al., 2023b; Peng et al., 2024). Coreference resolution typically only involves the replacement of a single span (most of them are entities, e.g., *"Ben Affleck"*), while ellipsis resolution might insert multiple discontinuous spans (e.g., *"as"* and *"Batman"*) from dialogue history at the same position in an incomplete utterance. Therefore, previous work treated coreference and ellipsis as a whole, which failed to account for the fundamental distinction between coreference and ellipsis. This resulted in the generation of utterances containing erroneous tokens or structures.

To address the aforementioned issues, we draw inspiration from the process of human article editing. When editing an article, humans typically read the entire text first, identify each area that requires insertion, deletion or replacement, and then reread the entire article to assess the reasonableness of each change. This prompts the question of *whether it is possible to explicitly introduce editing operations into incomplete utterance rewriting.*

We instantiate this idea by adopting a similar pipeline framework with a generation model to simulate human article editing. Specifically, we propose a novel framework called TEO (*Two-stage*

* Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 496–507
January 19–24, 2025.

approach on Editing Operation). We first use the editing operations as the pivot in the rewriting process, and train a model to generate editing operations in the first stage. The second stage is responsible for generating rewritten utterances based on the dialogue context and the editing operations. Furthermore, to mitigate the exposure bias caused by inconsistency between training and inference in the second stage, we propose three perturbation methods of editing operations to improve robustness. The experimental results on three IUR datasets show that our TEO significantly outperforms the SOTA models. In summary, our two-stage generation framework TEO has three advantages:

- The editing operations are taken as the pivot, with both insertion and replacement operations explicitly considered in order to address the issues of coreference and ellipsis.
- The local editing operations are first generated, and then the global dialogue context information and the editing operations are used to generate the final rewritten utterances. This enables the TEO to capture more fine-grained information.
- An adversarial perturbation strategy is proposed for editing operations that can mitigate the occurrence of cascading errors and exposure bias caused by inconsistency between training and inference.

2 Related Work

Research on IUR can be mainly divided into two types: generation methods (Huang et al., 2021; Inoue et al., 2022; Li et al., 2023b) and sequence labeling methods (Liu et al., 2020; Jin et al., 2022; Si et al., 2022; Chen, 2023; Du et al., 2023; Li et al., 2023a; Peng et al., 2024).

Most previous studies did not explicitly consider coreference resolution and ellipsis resolution. The initial research on IUR predominantly employed generation methodologies. For example, Huang et al. (2021) first employed a tagger to predict the rewriting labels and then utilized an autoregressive with copy mechanism for generating the rewritten utterances. Inoue et al. (2022) jointly trained two tasks: selecting key words and generating rewritten utterances.

Subsequent studies have indicated that the source and target utterances exhibit similar structural characteristics. Consequently, numerous subsequent studies have proposed sequence labelling methods. For example, Liu et al. (2020) regarded incomplete utterance rewriting as predicting word-level editing matrix. To address the issue of inserting multiple spans at one location, Chen (2023) directly selected spans from the context to form complete utterances. Du et al. (2023) incorporated sentence-level semantic relations between dialogue context and incomplete utterance. Li et al. (2023a) introduced the MLP architecture to mine the correlation between the contextual utterances and the rewritten utterance to obtain the editing matrix. Peng et al. (2024) paired spans and labeling the action types between spans.

Only a few works on IUR (Si et al., 2022; Li et al., 2023b) explicitly considered coreference and ellipsis resolution. Si et al. (2022) inserted markers into incomplete utterances to represent coreference and ellipsis through manually designed rules. However, these rules cannot cover all situations. Li et al. (2023b) first predicted the positions of insertion and replacement, and then filled these positions. However, they replaced the spans to be replaced with `[MASK]`, resulting in the loss of span information before replacement.

3 Methodology

3.1 Task Definition

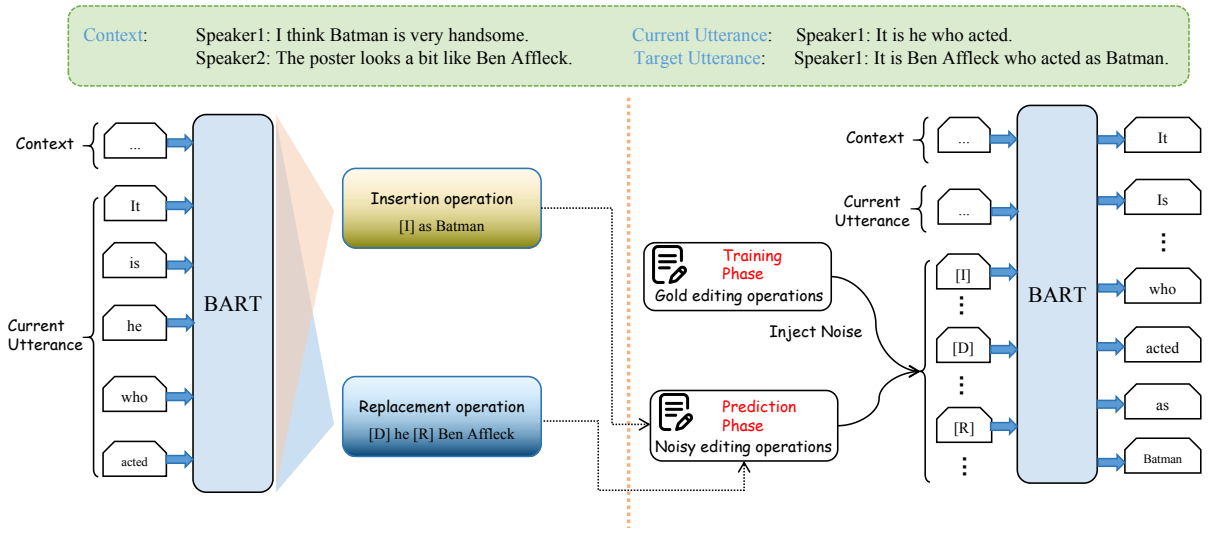
Let each piece of data be defined as $\{Hist, U_n, Y\}$, where $Hist = \{U_1, U_2, ...U_{n-1}\}$ is the dialogue history including $n-1$ utterances and U_n is the incomplete utterance that requires rewriting. The rewritten utterance of U_n is denoted as Y, where Y keeps the semantics of U_n unchanged and complements the coreferential and omitted information in it. The goal of IUR is to learn a mapping $P(T|Hist, U_n)$ satisfying $Y = \arg\max_T P(T|Hist, U_n)$.

3.2 Overview

We adopt a text generation approach to the IUR task. In contrast to previous research, which has treated IUR as a single task, we decompose it into two subtasks: editing operation generation and editing-aware rewritten utterance generation by using the editing operations $\mathcal{E}$ as the pivot. The objective of the overall training process is to maximize

$$P(Y|Hist, U_n) = P(\mathcal{E}|Hist, U_n)P(Y|Hist, U_n, \mathcal{E}), \quad (1)$$

with the goal of maximizing $P(\mathcal{E}|Hist, U_n)$ in the first stage and maximizing $P(Y|Hist, U_n, \mathcal{E})$ in



(a) The first stage: Editing Operation Generation

(b) The second stage: Editing-aware Rewritten Utterance Generation

Figure 1: An overview of our framework TEO, which includes two stages Editing Operation Generation and Editing-aware Rewritten Utterance Generation.

the second stage, where $\mathcal{E}$ is the predicted editing operations in the first stage.

Our framework TEO is shown in Figure 1. In the first stage *Editing Operation Generation*, we generate the editing operations based on the dialogue history and the incomplete utterance that needs to be rewritten. Subsequently, in the second stage *Editing-aware Rewritten Utterance Generation*, we generate the rewritten utterance based on the editing operations and dialogue context.

3.3 Editing Operation Generation

In the process of text editing, humans primarily engage in the insertion, deletion, and replacement of tokens. In the first stage, the objective is to generate editing operations to guide IUR. As previously stated, IUR primarily aims to address the coreference and ellipsis issues in utterances. The coreference resolution is typically achieved through replacement operations, while the ellipsis resolution is typically achieved through insertion operations. Consequently, two editing operations have been defined for IUR: insertion and replacement.

To express the semantics of the editing operations in a more concise manner, it is not necessary to use natural language descriptions to represent insertion and replacement operations. Instead, three types of markers are employed for insertion and replacement. Specifically, we use "$[I]$ tk" to represent an insertion operation where "$[I]$" refers to the insertion action and "tk" refers to the inserted tokens. The replacement operation is indicated by the following sequence: "$[D]$ tk_1 $[R]$ tk_2". Here, "$[D]$" and "$[R]$" refer to the deletion and replacement action, respectively, and the tokens "tk_1" are replaced by "tk_2".

By doing so, we can obtain the structured editing operations and denote it as $\mathcal{E}$, where each of the editing operations is ordered according to its position in the original utterance. As shown in Figure 1, by comparing the incomplete utterance *"It is he who acted"* with the rewritten utterance *"It is Ben Affleck who acted as Batman"*, we can obtain the set of editing operations as "*[D] he [R] Ben Affleck [I] as Batman*".

To train a generation model to generate editing operations, we use the incomplete utterances and the rewritten utterances in the training set to create editing operations. Specifically, we first compare the incomplete utterance U_n and the rewritten utterance Y to find their longest common subsequence, denoting as $\mathcal{L}$. For those tokens that appear in U_n but not in $\mathcal{L}$, we mark them as *deletion*. For those tokens that appear in Y but not in $\mathcal{L}$, we mark them as *insertion*. Finally, we obtain a set of deletion and insertion operations. For each deletion and insertion operation with the same context, we label it as *replacement*, resulting in a set of replacement operations. By using the above methods, we can obtain the insertion operation set $\mathcal{I}$ and the replacement operation set $\mathcal{R}$. For each incomplete utterance, our goal of editing operation generation is to generate the set of editing operations as follows,

$$T(\mathcal{I},\mathcal{R}) = \text{[I]}I_1 ... \text{[I]}I_i ... \text{[D]}D_1\text{[R]}R_1 ... \text{[D]}D_j\text{[R]}R_j, \quad (2)$$

where I_i represents the i-th span to be inserted, and

D_j and R_j represent the j-th span to be replaced and the span after replacement, respectively.

We use BART (Lewis et al., 2020) as our generation model and concatenate the dialogue history $Hist$ and the incomplete utterance U_n as input. The output is the set of editing operations $\mathcal{E}$ of length l. The training objective of the editing operation generation is to minimize the negative log-likelihood loss function $Loss_{edit}$ as follows,

$$Loss_{edit} = -\log \sum_{i=1}^{l} P(\mathcal{E}_i|\mathcal{E}_{<i}, Hist, U_n, \theta_1), \quad (3)$$

where θ_1 is the parameters of the model in the first stage.

3.4 Editing-aware Rewritten Utterance Generation

Our editing process in the first stage can be seen as a process of proposing insertions and replacements. Due to the errors in the predicted editing operations of the first stage and the unknown positions for the insertion operations, we combine global dialogue information with the preliminary generated editing operations to generate the final rewritten utterances in the second stage. We hope that the model can review and refine the generated editing operations based on contextual information, in order to obtain correct rewritten utterances. Specifically, we train the model to generate the final rewritten utterances based on the dialogue history, the utterances to be rewritten, and the editing operations. Moreover, to construct editing-aware context data, we use the correct editing operations during the training stage and use the noisy editing operations generated by the first stage during the prediction stage.

The exposure bias caused by the inconsistency in the training and inference stages can affect the effectiveness of the model, leading to overfitting of correct editing operations. Here we adopt three editing operation perturbation methods to alleviate exposure bias. Specifically, we add three types of perturbations to the editing operations during training to improve the robustness of the model, namely random replacement, random deletion and random insertion.

The probability of perturbations is denoted by $prob_p$. For gold editing operations, the probability of randomly replacing the span of the operation is $prob_r$, while the probability of randomly deleting it is $1 - prob_r$. For each editing operation e, we first sample two probability values $prob_1$ and $prob_2$ from a uniform distribution. If $prob_1 \leq prob_p$, we perform random deletion or random replacement as follows: if $prob_2 \leq prob_r$, we randomly replace the text (corresponding span) with a random span sampled from dialogue (Line 6-9); otherwise, we delete this editing operation (i.e., do not insert into E_p). Taking the editing operation "*[D] he [R] Ben Affleck*" as an example, we use "*Batman*" to replace "*Ben Affleck*" and form a new editing operation "*[D] he [R] Batman*". Then, we randomly sample a probability value $prob_3$ from a uniform distribution. if $prob_3 \leq prob_p$, we perform a random insertion (Line 15-20), including an insertion or a replacement operation. Taking the Figure 1 as an example, we insert an insertion operation "*[I] Batman*" or a replacement operation "*[D] He [R] Batman*" to E_p, resulting in an adversarial sample.

Algorithm 1: Perturbation Strategy

```
input : Editing operations 𝒯; Perturbation
        probability prob_p; Span replacement
        probability prob_r
output: Perturbed editing operations E_p
1  E_p ← {};
2  foreach e in 𝒯 do
3      Draw prob_1, prob_2 from Uniform(0, 1);
4      // Add noise to editing operations with
          a probability of prob_p
5      if prob_1 <= prob_p then
6          if prob_2 <= prob_r then
7              e.text ← A random span sampled
                from the dialogue history;
8              E_p ← E_p ∪ {e}
9          end
10     else
11         E_p ← E_p ∪ {e}
12     end
13 end
14 Draw prob_3 from Uniform(0, 1);
15 if prob_3 <= prob_p then
16     origin_span ← A random span sampled from the
         incomplete utterance;
17     new_span ← A random span sampled from the
         dialogue history;
18     candidates=["[I] new_span", "[D] origin_span
         [R] new_span"];
19     E_p ← E_p∪ {the operation randomly sampled
         from the candidates}
20 end
21 return E_p
```

We also use BART as the generation model and concatenate the history $Hist$, the incomplete utterance U_n and the editing operations $\mathcal{E}$ as input, i.e., $\{\texttt{[CLS]}Hist\texttt{[SEP]}U_n\texttt{[SEP]}\mathcal{E}\texttt{[SEP]}\}$, where [CLS] and [SEP] represent the special tokens in BART. The output is the rewritten utterance $Y = \{y_1, ...y_i, ...y_m\}$ where y_i is the i-th token. The training objective of the editing-aware rewritten

Model	EM	B_4	F_1
$BART_{base}$ (Lewis et al., 2020)	70.1	83.9	69.5
QUEEN (Si et al., 2022)	71.6	86.3	NA
SGT (Chen, 2023)	71.1	86.7	85.0
MIUR (Li et al., 2023a)	70.9	86.0	72.3
Locate-Fill (Li et al., 2023b)	75.0	87.3	84.2
XSS (Peng et al., 2024)	70.2	85.6	70.4
TEO-Stage1	52.7	75.7	62.5
TEO-RFIS	77.7	88.5	85.3
TEO-T5	76.4	87.6	84.5
TEO (Ours)	**78.1**	**88.7**	**85.8**
TEO-Gold	83.5	91.5	90.9

Table 2: Result comparison on English TASK.

utterance generation is to minimize the negative log-likelihood loss function $Loss_{uttr}$ as follows,

$$Loss_{uttr} = -\log \sum_{i=1}^{m} P(y_i|y_{<i}, Hist, U_n, \mathcal{E}, \theta_2), \quad (4)$$

where θ_2 is the parameters of the model in the second stage. We initialize the model using the weights of the model trained in the first stage.

4 Experimentation

4.1 Experimental Settings

Datasets We conduct experiments on three popular IUR datasets: the Chinese open-domain dialogue datasets REWRITE (Su et al., 2019) and RESTORATION-200K (abbreviated as RES200K) (Pan et al., 2019), and the English task-oriented dataset TASK (Quan et al., 2019). Specific statistics for the datasets are provided in Appendix A.

Metrics In order to evaluate our method, we employ four different automatic evaluation metrics, namely EM, $BLEU_n$ (abbreviated as B_n) (Papineni et al., 2002), $ROUGE_n$ (R_n) (Lin, 2004), and Restoration F-score$_n$ (F_n) (Pan et al., 2019). These metrics can effectively reflect the quality of rewriting. Following previous work (Liu et al., 2020; Si et al., 2022; Li et al., 2023a), we also report different metrics on different datasets.

Baselines We compare our TEO with the following strong baselines: $BART_{base}$ (Lewis et al., 2020), QUEEN (Si et al., 2022), RAU (Zhang et al., 2022), SGT (Chen, 2023), MIUR (Li et al., 2023a), Locate-Fill (Li et al., 2023b), MGIIF (Du et al., 2023) and XSS (Peng et al., 2024).

The implementation and details of metrics and baselines are provided in Appendix B, C and D.

4.2 Experimental Results

We evaluate our TEO and the baselines on the three datasets and the results are shown in Tables 2, 3 and 4, respectively, where "NA" refers to the metrics that the original paper did not report. The results on three datasets show that our TEO outperforms the baselines on almost all metrics. In terms of the most rigorous metric, EM, our TEO achieves an improvement of 3.1, 2.6 and 0.6 on the TASK, REWRITE and RES200K datasets, respectively, in comparison with the previous SOTA models. This result demonstrates that the robustness of our TEO allows it to avoid various minor boundary errors. Furthermore, the improvements observed on the English TASK and Chinese REWRITE and RES200K datasets demonstrate the applicability of our TEO to diverse languages and domains.

Furthermore, it is observed that on the three datasets, our TEO exhibits superior performance compared to the baselines in F_n, while it is almost on par with them in $BLEU_n$. This can be attributed to the fact that F_n emphasizes the ability of the model to identify omitted or referential information in the incomplete utterance, whereas $BLEU_n$ considers all n-gram information present in the utterance including the span that is already present in the original incomplete utterance. The higher performance in F_n indicates that our TEO can better capture contextual information of conversations and subsequently address the issues of coreference and ellipsis based on the context. Moreover, we find that our TEO achieves the improvements of 0.8, 2.2 and 2.6 in F_1, F_2, and F_3 scores, respectively, in comparison with the previous SOTA model MIUR on the RES200K dataset. In essence, the F_1 score is indicative of the accuracy of token restoration, irrespective of the insertion position within the utterance. However, the positioning of newly added tokens will influence the values of F_2 and F_3. The enhanced performance observed in F_2 and F_3 also suggests that our TEO is capable of not only generating missing tokens but also inserting them at the appropriate locations.

To ensure a fair comparison, we maintained consistent experimental settings with previous related studies and utilized $BART_{base}$ as the backbone. Additionally, we conducted experiments on three datasets using $T5_{base}$ as the backbone and the results of these experiments are presented in Tables 2, 3 and 4, as TEO-T5, respectively. It is worth noting that $BART_{base}$ outperformed $T5_{base}$ in most

Model	F_1	F_2	F_3	B_1	B_2	R_1	R_2	R_L	EM
$BART_{base}$ (Lewis et al., 2020)	81.2	76.0	79.7	93.9	90.8	95.2	91.8	92.4	70.5
RAU (Zhang et al., 2022)	NA	NA	NA	NA	91.6	NA	90.6	93.9	68.4
QUEEN (Si et al., 2022)	NA	NA	NA	NA	92.1	NA	90.9	94.6	70.1
MIUR (Li et al., 2023a)	NA	82.2	NA	NA	91.2	NA	90.7	93.7	67.7
Locate-Fill (Li et al., 2023b)	89.9	83.9	79.4	93.8	91.8	95.9	91.6	94.0	70.9
XSS (Peng et al., 2024)	89.8	82.0	76.1	92.4	91.0	95.8	90.7	93.7	66.7
TEO-Stage1	85.6	63.0	28.3	90.9	82.6	94.0	75.9	75.9	34.0
TEO-RFIS	90.0	83.9	80.9	93.3	91.2	95.9	91.6	94.2	72.1
TEO-T5	**91.2**	**85.9**	81.1	94.1	92.2	96.2	92.0	94.1	72.6
TEO (Ours)	91.0	85.4	**82.1**	**94.4**	**92.5**	**96.3**	**92.3**	**94.7**	**73.5**
TEO-Gold	98.1	94.3	91.2	98.4	97.3	99.1	96.7	97.5	86.3

Table 3: Result comparison on Chinese REWRITE.

Model	P_1	R_1	F_1	P_2	R_2	F_2	P_3	R_3	F_3	B_1	B_2	R_1	R_2	EM
$BART_{base}$ (Lewis et al., 2020)	70.9	55.8	62.4	60.8	47.4	53.3	54.0	41.8	47.1	90.5	87.9	91.8	85.5	52.9
RAU (Zhang et al., 2022)	75.0	65.5	69.9	61.2	54.3	57.5	52.5	47.0	49.6	92.4	89.6	92.8	86.0	NA
QUEEN (Si et al., 2022)	NA	NA	NA	NA	NA	NA	NA	NA	NA	92.4	89.8	92.5	86.3	53.5
MGIIF (Du et al., 2023)	NA	NA	70.8	NA	NA	58.5	NA	NA	50.5	**93.1**	**90.4**	93.2	86.6	NA
MIUR (Li et al., 2023a)	76.4	63.7	69.5	62.7	52.7	57.3	54.3	45.9	49.7	93.0	90.1	92.6	85.7	51.0
Locate-Fill (Li et al., 2023b)	73.1	61.9	67.0	62.6	52.4	57.0	55.4	46.0	50.2	92.5	89.9	92.5	86.3	53.6
XSS (Peng et al., 2024)	NA	NA	**70.9**	NA	NA	57.0	NA	NA	47.9	92.5	89.7	92.7	85.9	50.1
TEO-Stage1	73.8	65.0	69.1	42.7	39.5	41.0	24.6	23.0	23.7	92.9	86.8	92.6	78.6	42.5
TEO-RFIS	74.2	66.6	70.2	62.7	55.9	59.1	55.0	48.9	51.8	92.5	89.7	92.7	86.2	53.3
TEO-T5	**76.7**	65.6	70.7	61.4	55.1	58.1	53.8	48.3	50.9	91.3	88.7	**93.6**	**87.9**	53.6
TEO (Ours)	74.4	**66.7**	70.3	**63.4**	**56.1**	**59.5**	**55.9**	**49.1**	**52.3**	92.8	90.1	92.8	86.5	**54.2**
TEO-Gold	93.5	94.3	93.9	83.9	84.1	84.0	76.8	76.9	76.9	97.6	95.6	98.0	93.6	80.3

Table 4: Result comparison on Chinese RES200K.

metrics, which may be attributed to the different pre-training objectives of the two models. During pre-training, BART introduces noise to the text and reconstructs the original text at the decoder. In contrast, T5 models various classification and generation tasks in a unified text-to-text format during pre-training. BART's pre-training objective is similar to our IUR task because coreference and ellipsis in IUR can be viewed as a type of noise that our task aims to recover.

4.3 Analysis on Editing Operation Generation

As shown in Table 5, we calculate the EM metric of the editing operations generated in the first stage, i.e., editing operation generation. It can be observed that the EM metric is also relatively high. For example, in the TASK and REWRITE datasets, the EM metrics of the first stage are 73.1 and 75.2, respectively, while the metrics for the second stage are 78.1 and 73.5, respectively. It can be observed that the EM metric is higher in the second stage of the TASK dataset in comparison with the first stage. This further corroborates the hypothesis that even if erroneous editing operations are generated in the first stage, a portion of them can be rectified in the second stage.

To further validate the effectiveness of the first stage, we use the correct editing operations during inference in the second stage, shown in Tables 2, 3 and 4, as TEO-Gold, respectively. The results show a significant improvement in all metrics for all three datasets after using the correct editing operations. This indicates that the performance of editing operation generation is positively correlated with that of the second stage. Simultaneously, the performance of directly using $BART_{base}$, i.e., removing the first stage, is much lower than our TEO, which also proves the effectiveness of the first stage and illustrates that the editing operations are the pivot for the IUR task.

4.4 Analysis on Editing-aware Rewritten Utterance Generation

To verify the effectiveness of the second stage, i.e., editing-aware rewritten utterance generation, it is not necessary to proceed to the second stage after completing the first stage. Instead, the rewritten utterance can be generated directly according to the following rules: For those replacement operations, we directly parse the editing operations generated

Dataset	EM	E2C(%)	C2E(%)
TASK	73.1	9.71	20.56
REWRITE	75.2	12.53	3.41
RES200K	59.8	4.82	29.66

Table 5: The EM, E2C and C2E of the first stage.

in the first stage. For insertion operations, since it is not possible to determine the exact positions at which the insertion should occur, a random selection of positions is made. The results are shown in Tables 2, 3 and 4, as TEO-Stage1 respectively. It can be observed that the outcomes of the first stage are considerably inferior to those of the two-stage method TEO. This evidence corroborates the efficacy of the second stage and the beneficial interaction between the first and second stages.

In addition, we define two metrics, error to correct rate ($E2C$) and correct to error rate ($C2E$),

$$E2C = \frac{\#err\_cor}{\#er}, C2E = \frac{\#cor\_err}{\#cor}, \quad (5)$$

where $\#err\_cor$ refers to the number of samples that were predicted incorrectly in the first stage but correctly in the second stage, $\#er$ refers to the number of samples that were predicted incorrectly in the first stage, $\#cor\_err$ refers to the number of samples that were predicted correctly in the first stage but incorrectly in the second stage, and $\#cor$ refers to the number of samples that were predicted correctly in the first stage. The first metric measures the proportion of samples that were incorrectly predicted in the first stage but correctly predicted in the second stage. The second metric measures the proportion of samples that were correctly predicted in the first stage but incorrectly predicted in the second stage. With these two metrics, we can quantitatively analyze the correlation between the two stages.

As shown in Table 5, we find that $E2C$ is relatively higher than $C2E$ on the REWRITE dataset. This indicates that a higher proportion of incorrect editing operations in the first stage are corrected in the second stage, thereby proving that the combination of global dialogue context information with local editing operation information is effective. Through our observations, we find that there are far more cases of ellipsis than the cases of coreference in RES200K and TASK. This implies that there are more instances of insertion. In the first stage, we only obtain the inserted tokens but do not know its positions. Therefore, even if we correctly predict the inserted tokens in the first stage, there

Context: $\mathbf{Speaker_1}$: Don't you have any musical instruments that you want to learn? I think the piano and guitar sound great. $\mathbf{Speaker_2}$: Piano. $\mathbf{Speaker_1}$: The sound of this musical instrument sounds very pleasant, wow. (**Incomplete utterance**)
Reference: The sound of the piano sounds very pleasant, wow.
Correct editing operation: [D] this musical instrument [R] the piano
Predicted editing operation in the first stage: [D] this [R] the piano ✗
Predicted rewritten utterance in the second stage: The sound of the piano sounds very pleasant, wow. ✓

Table 6: A boundary error in the first stage is corrected by the second stage.

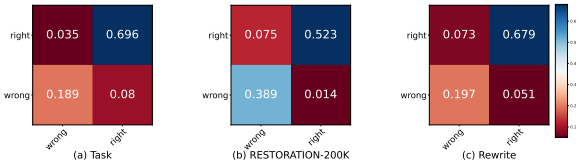


Figure 2: The correspondence between the positive and negative examples in the first stage and the second stage, where "wrong" and "right" on the vertical axis respectively represent incorrect and correct editing operations in the first stage, while "wrong" and "right" on the horizontal axis respectively represent incorrect and correct rewritten utterances in the second stage.

are still many cases of incorrect insertion positions in the second stage. Consequently, the value of $C2E$ is higher in RES200K and TASK.

Since the first stage can solve the issue of coreference, the replacement operations can be used to rewrite utterances. Instead of that, we feed both the replacement and insertion operations to the second stage. To conduct insightful analysis, we perform predictive replacement operations after the first stage and only handle insertion operations in the second stage. The results are shown in Tables 2, 3, 4 as TEO-RFIS, respectively. This approach yields inferior results across all metrics compared to simultaneously handling replacement and insertion operations in the second stage. Taking table 6 as example, the first stage generates an incorrect replacement operation, predicting the replacement of "this musical instrument" with "this". If the replacement is executed directly, an incorrect output is produced. However, the second stage is able to correct this error and produce the correct utterance.

Furthermore, we also investigated the results predicted in the second stage for the samples corresponding to correct or incorrect editing operations

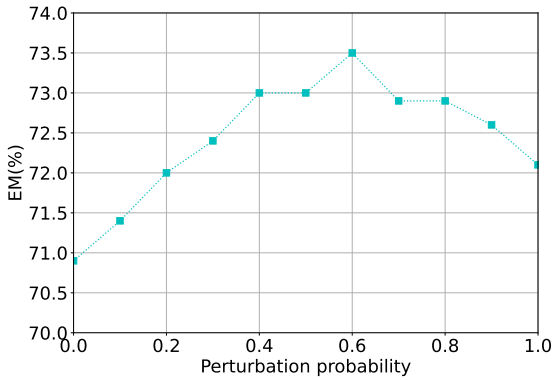


Figure 3: The trend chart of EM metric changing with the variation of adversarial perturbation probability on REWRITE.

predicted in the first stage. The results are shown in Figure 2. The majority of samples with accurate editing operation predictions in the first stage were correctly identified in the second stage, while some of the incorrectly predicted samples were mitigated in the second stage. We find that 8.0%, 5.1% and 1.4% of samples are corrected by the second stage on TASK, REWRITE, and RES200K respectively. However, these samples did not have correct editing operations in the first stage. This suggests that our second stage editing-aware rewritten utterance generation is capable of correcting the errors produced in the first stage.

It is worth noting that the edit operations predicted in the first stage can be used in the training of the second stage and our experimental results show the values of F_1 and EM were 89.7 and 72.3, respectively, which are inferior to our TEO (91.2 and 73.5). This is because it would result in a fixed distribution of erroneous editing operations in the training data of the second stage. Nevertheless, the utilization of our proposed adversarial perturbation strategy enables the dynamic adjustment of noise within training samples, thereby enhancing the model's robustness.

4.5 Impact of Adversarial Perturbation

As mentioned in Section 3.4, the introduction of adversarial perturbations is employed to mitigate the impact of exposure bias, which arises due to inconsistency between the training and prediction stages. To assess the efficacy of adversarial perturbations, experiments were conducted with varying perturbation probabilities, and the experimental results are presented in Figure 3. As the probability of perturbation increases, the EM value initially increases and then decreases, reaching its peak when

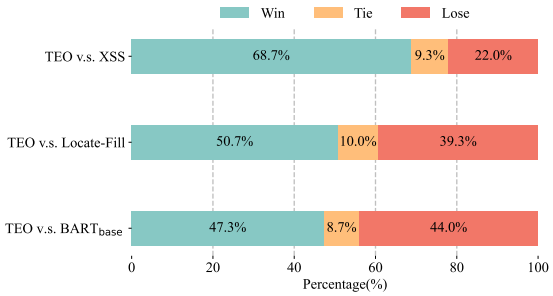


Figure 4: Performance comparison of our method with $BART_{base}$, Locate-Fill and XSS on human evaluation.

the probability is 0.6. This may be due to the inconsistency between the training and inference data distributions when the perturbation probability is low. When the perturbation probability is too high, TEO fails to capture knowledge in editing operations. However, we observe that even when the perturbation probability is 1, TEO can still achieve good results. This also indicates that in situations that are close to zero-shot, our TEO can still perform well.

4.6 Human Evaluation

Incomplete utterance rewriting often results in non-unique rewriting outcomes, making it challenging to fully assess the performance of the model based on automatic evaluation metrics alone. To address this, we conducted a human evaluation to compare our proposed method with $BART_{base}$, Locate-Fill and XSS. A total of fifty data points were randomly sampled from the test set and distributed to three raters. Each rater selected a better result from the outputs generated by two methods.

The results are presented in Figure 4, it can be seen that our method outperforms $BART_{base}$, Locate-Fill and XSS, indicating consistency with the results observed in our automatic evaluation. However, it was observed that our method has a smaller advantage when evaluated manually against $BART_{base}$, compared to Locate-Fill and XSS. After analyzing the rewritten utterances produced by these four methods, it was discovered that while $BART_{base}$ demonstrated an average ability to rewrite, it was capable of generating more fluent utterances, which ultimately led to superior human evaluation results.

4.7 Comparison with ChatGPT

At present, the majority of LLMs lack the capacity to process coreference and ellipsis resolution, which are integral to IUR. We conducted prelim-

Model	F_1	F_2	B_1	B_2	R_1	R_2	R_L	EM
GPT-4	79.1	66.7	85.3	80.4	87.7	77.6	82.1	35.7
Ours	**91.0**	**85.4**	**94.4**	**92.5**	**96.3**	**92.3**	**94.7**	**73.5**

Table 7: Performance comparison between TEO and GPT-4 on REWRITE.

inary experiments utilising the in-context learning method with GPT-4 (the version used here is `gpt-4-1106-preview`) as a case study and the prompt used in GPT-4 is provided in Appendix E.

We provided five samples as demonstrations to GPT-4 and the experimental results on REWRITE are shown in Table 7. Even though the number of parameters in GPT-4 is much higher than our model, our model still achieves better performance than it. In addition, we observed that some utterances generated by GPT-4 do not match incomplete utterances in terms of semantics, which also indicates that the understanding of global conversational semantics by LLMs needs to be improved.

4.8 Case Study

Our TEO is capable of accommodating both instances where the rewritten utterance contains tokens that are not present in the dialogue history, as well as instances where multiple discontinuous spans must be inserted at the same position in the current utterance. We compare our TEO with three baselines, i.e., $\text{BART}_{\text{base}}$, RUN, and HCT, and the results on the Chinese REWRITE dataset are shown in Table 8. We need to insert four spans "天龙八部里" (in Demi-Gods and Semi-Devils), "段誉" (Duan Yu), "的" (of), and "武功最高" (the highest martial arts skill) at the end of the current utterance. Although $\text{BART}_{\text{base}}$ generates a more fluent rewritten utterance, it does not fit into the context of the conversation. RUN selects all correct spans but fails to insert them in the correct order within the utterance (Although the English translation is coherent, there is an word order error in Chinese sentences). HCT has boundary errors for some spans; for example, instead of inserting "段誉" (Duan Yu), it inserts "是段誉" (is Duan Yu). Additionally, there is also an issue with span insertion order. Only our TEO outputs correct and complete utterance, which benefits from the complementary effects of the two-stage mechanism.

4.9 Error Analysis

To analyse the errors in our TEO model, we compiled the experimental results and found that the majority of errors arise from the insertion operation rather than the replacement operation. For instance, in REWRITE, insertion errors account for 79.3%, while replacement errors account for only 20.7%. This outcome is mainly due to the uncertainty of the insertion position. Secondly, in contrast to REWRITE, RES200K and TASK contain utterances that do not require rewriting. This can lead to errors when editing these utterances. For example, in RES200K, 38.7% of utterances do not require rewriting and have an EM score of 77.5, with almost all errors resulting from incorrect replacement operations. Finally, the EM scores for the replacement operation are considerably lower for RES200K (0) and TASK (28.6) than for REWRITE (83.0). This is primarily due to the limited number of replacement operations in these two datasets (RES200K: 0.15%; TASK: 12.42%).

Context: *A*: 天龙八部里 谁 的 武功最高 (Who has the highest martial arts skill in "Demi-Gods and Semi-Devils"?) *B*: 是 段誉 (It's Duan Yu.) *A*: 为什么(Why?) (**Incomplete utterance**)
Reference: 为什么 天龙八部里 段誉 的 武功最高 (Why is Duan Yu's martial arts the highest in "Demi-Gods and Semi-Devils"?)
$\text{BART}_{\text{base}}$: 为什么天龙八部里谁的武功最高(Why is whose martial arts the highest in "Demi-Gods and Semi-Devils"?) ✗
RUN: 为什么天龙八部里的武功最高段誉(Why is the martial arts in "Demi-Gods and Semi-Devils" the highest, Duan Yu?) ✗
HCT: 为什么天龙八部里谁武功最高是段誉(Why is Duan Yu whose martial arts is the highest in "Demi-Gods and Semi-Devils"?) ✗
TEO (Ours): 为什么天龙八部里段誉的武功最高(Why is Duan Yu's martial arts the highest in "Demi-Gods and Semi-Devils"?) ✓

Table 8: A case of our TEO and three baselines $\text{BART}_{\text{base}}$, RUN and HCT on REWRITE.

5 Conclusion

We propose a two-stage IUR framework by taking the editing operations as the pivot, in which the first stage generates editing operations for IUR and the second stage rewrites incomplete utterances utilizing the generated editing operations and the dialogue context. Moreover, an adversarial perturbation strategy is proposed to enhance model robustness. The experimental results on three IUR datasets show that our TEO outperforms the SOTA models significantly. Our future work will focus on how to introduce LLMs to assist IUR.

Limitations

Although this paper may contribute to incomplete utterance rewriting and some downstream dialogue tasks, it still suffers from two shortcomings, which are our future work. First, we only use one representation of the editing operation in our model. We believe that better templates can help the model better understand and improve the effectiveness of dialogue rewriting. Second, we only used the editing operations generated in the first stage to assist in rewriting the dialogue utterances in the second stage, but did not attempt to use the dialogue rewriting of the second stage to facilitate the first stage. Therefore, how to promote the complementary interaction between the two stages is our future research.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (Nos. 62376181 and 62276177), and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Yunshan Chen. 2023. Incomplete utterance rewriting as sequential greedy tagging. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7265–7276.

Haowei Du, Dinghao Zhang, Chen Li, Yang Li, and Dongyan Zhao. 2023. Multi-granularity information interaction framework for incomplete utterance rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2576–2581.

Yue Fang, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Bo Long, Yanyan Lan, and Yanquan Zhou. 2022. From spoken dialogue to formal summary: An utterance rewriting for dialogue summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3859–3869.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021. RAST: Domain-robust dialogue rewriting as sequence tagging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4913–4924.

Mengzuo Huang, Feng Li, Wuhe Zou, and Weidong Zhang. 2021. SARG: A novel semi autoregressive generator for multi-turn incomplete utterance restoration. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021*, pages 13055–13063.

Shumpei Inoue, Tsungwei Liu, Son Nguyen, and Minh-Tien Nguyen. 2022. Enhance incomplete utterance restoration by joint learning token extraction and text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3149–3158.

Lisa Jin, Linfeng Song, Lifeng Jin, Dong Yu, and Daniel Gildea. 2022. Hierarchical context tagging for utterance rewriting. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022*, pages 10849–10857.

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.

Jiang Li, Xiangdong Su, Xinlan Ma, and Guanglai Gao. 2023a. How well apply simple MLP to incomplete utterance rewriting? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1567–1576.

Zitong Li, Jiawei Li, Haifeng Tang, Kenny Zhu, and Ruolan Yang. 2023b. Incomplete utterance rewriting by a two-phase locate-and-fill regime. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2731–2745.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2846–2857.

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

Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1824–1833.

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.

Letian Peng, Zuchao Li, and Hai Zhao. 2024. Fast and accurate incomplete utterance rewriting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–11.

Hongjin Qian and Zhicheng Dou. 2022. Explicit query rewriting for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4725–4737.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4547–4557.

Shuzheng Si, Shuang Zeng, and Baobao Chang. 2022. Mining clues from incomplete utterance: A query-enhanced network for incomplete utterance rewriting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4839–4847.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31.

Yong Zhang, Zhitao Li, Jianzong Wang, Ning Cheng, and Jing Xiao. 2022. Self-attention for incomplete utterance rewriting. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore*, pages 8047–8051.

A Dataset Statistics

The specific information of the three datasets is shown in Table 9.

Category	REWRITE	RES200K	TASK
Language	Chinese	Chinese	English
Train	18K	194K	2.2K
Dev	2K	5K	0.5K
Test	2K	5K	0.5K
#Avg. Cont	17.7	25.5	52.6
#Avg. Curr	6.5	8.6	9.4
#Avg. Rewr	10.5	12.4	11.3
#Insertion	14070	136339	1572
#Replacement	7853	203	223

Table 9: Statistics of different datasets, where "Cont", "Curr" and "Rewr" are the abbreviations for the context, current, and rewritten utterance, respectively.

B Implementation

The pre-trained $\text{BART}_{\text{base}}$ is employed as the backbone model, with all experiments conducted using the open-source library PyTorch. To reduce the latency during inference, a greedy decoding strategy is employed. Both the first stage and the second stage are fine-tuned for 30 epochs. The AdamW optimiser (Loshchilov and Hutter, 2019) is used, with a learning rate of 5e-5. In the second stage, the probabilities of random deletion and random replacement are both set to 0.5. Given that the length of the edit operation is considerably shorter than that of the rewritten utterance, the decoding time of the first stage is negligible in comparison to that of the second stage.

C Details of Metrics

EM refers to exact matching accuracy, which is a strict metric, representing the ratio of correctly predicted samples to the total number of samples. The BLEU metric evaluates accuracy by calculating the matching degree of n-grams. BLEU_1 is a metric that measures the accuracy at the word level, while higher-order BLEU can be used to assess the fluency of utterances. Additionally, we employ ROUGE_n to measure recall in IUR. ROUGE evaluates recall by counting n-gram co-occurrences. F_n (Pan et al., 2019) is utilized to identify the words that have been added to the utterance for rewriting. The n-gram restoration precision, recall and F-score are calculated as follows,

$$p_n = \frac{|\{\text{ restored n-grams }\} \cap \{\text{ n-grams in ref }\}|}{|\{\text{ restored n-grams }\}|}$$

$$r_n = \frac{|\{\text{ restored n-grams }\} \cap \{\text{ n-grams in ref }\}|}{|\{\text{ n-grams in ref }\}|}$$

$$f_n = 2 \cdot \frac{p_n \cdot r_n}{p_n + r_n}$$

where "restored n-grams" denotes n-grams in model-generated utterances that contain at least one restored word, and "n-grams in ref" denotes n-grams in reference utterances that contain at least one restored word.

D Details of Baselines

We introduce eight strong baselines to verify the effectiveness of our proposed model TEO as follows.

(1) $\text{BART}_{\text{base}}$: it generated rewritten utterance using dialog history and incomplete utterance as input;

(2) QUEEN (Si et al., 2022): it proposed a query template that was concatenated with utterance as input;

(3) RAU (Zhang et al., 2022): it extracted relations between tokens from a self-attention matrix;

(4) SGT (Chen, 2023): it first identified fragments and their relative order, and then generated the target utterance;

(5) MIUR (Li et al., 2023a): it mined latent semantic information through a layer of MLP and predicted token types through a joint feature matrix;

(6) Locate-Fill (Li et al., 2023b): it proposed a two-phase incomplete utterance rewriting method that first predicted empty slots and then filled them;

(7) MGIIF (Li et al., 2023b): it proposed a multi-task information interaction framework for incomplete utterance rewriting;

(8) XSS (Peng et al., 2024): it is an incomplete utterance rewriting model based on span pairing.

E Prompts Used in GPT-4 Evaluation

The prompt used in the GPT-4 evaluation is as follows.

Prompt used in GPT-4 assessment

The goal of dialogue rewriting is to resolve coreference and ellipsis, that is, to complete the coreferential and omitted information in the dialogue without changing its original semantics. Please rewrite the final utterance in the following dialogue.
Examples: `{Examples}`
Input: `{Input}`