Context-Aware Tracking and Dynamic Introduction for Incomplete Utterance Rewriting in Extended Multi-Turn Dialogues

Xinnan Guo\textsuperscript{1}, Qian Zhu\textsuperscript{1}, Qiuhui Shi\textsuperscript{1}, Xuan Lin\textsuperscript{1}, Liubin Wang\textsuperscript{1}, Daqian Li\textsuperscript{1}, Yongrui Chen\textsuperscript{1}

\textsuperscript{1}Ant Group, China
guoxinnan0727@163.com, qz371417, qiuhi.sqh, daxuan.lx, daqian.ldq@antgroup.com, ubiwang@gmail.com, yrchen@seu.edu.cn

Abstract

Incomplete utterance rewriting (IUR) aims to reconstruct the utterance with omitted information and pronouns to be standalone and complete based on the context. The existing works predominantly focus on simple ellipsis and coreference problems in brief multi-turn dialogues. But in actual scenarios: 1) the context of the dialogues frequently comprises multiple similar candidates for ellipsis and coreference resolution, pouring to confuse. 2) the number of turns tends to be more extensive, while the content with various topics also grows more complex. This paper proposes a novel method called CAT to address these issues. In particular, we first devise a tacker model, distilled from GPT4-turbo, to adopt Context Tracking that dynamically updates a list of key phrases turn by turn, as accurate candidates for ellipsis and coreference resolution. Second, we further present the Dynamic Context Introduction mechanism to filter irrelevant preceding contexts that are not relied on by any element within the key phrase list to condense extended dialogues. Comprehensive experiments indicate that our solution provides a significant improvement over the existing baselines, and achieves state-of-the-art on three benchmarks\textsuperscript{1}.

1 Introduction

Incomplete Utterance Rewriting (IUR) serves as a vital component in multi-turn dialogue systems, which reconstruct the current utterance by integrating the omitted information and resolving coreference based on the context to ensure its completeness. Currently, IUR is widely employed in prevalent tasks such as information retrieving (Li et al., 2022b; Mo et al., 2023), question answering (Vakulenko et al., 2021), and web searching (Li et al., 2022a; Mohankumar et al., 2023). With the advent of large language models (LLMs) (Ouyang et al., 2022; OpenAI, 2023) and breakthroughs in multi-turn dialogue-oriented tasks, IUR has received increasing attention in recent years.

In contrast to the brief (in less than 10 turns on average) and simple (usually centering around a single topic) dialogue observed between users and systems, which is primarily focused on by current IUR researches as the benchmark (Elgohary et al., 2019; Su et al., 2019; Pan et al., 2019; Regan et al., 2019; Martin et al., 2020), customer service and sales scenarios often entail more complex dialogues between users and salespersons, as shown in Figure 1. These scenarios present two principal challenges: a) complex ellipsis and coreference problems delivered by the diverse dialogue content and topics. For instance, according to the demand for reimbursement, several insurance products are mentioned throughout the dialogues, which may lead to confusion in figuring out which product

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example.png}
\caption{An example of IUR in a product consulting and sales scenario. Within, green denotes the user’s input, while blue signifies the salesperson’s. The full dialogue is upon 35 turns.}
\end{figure}

\footnotetext{1https://github.com/ygxw0909/CaT}
is the user inquiring about. b) extended turns, at times, reaching upwards of hundreds, encompassing multiple distinct topics. Within the dialogue in Figure 1, the topic transitions from consulting the function of medical insurance to recommending various products for purchase.

Some of the existing methods (Liu et al., 2020; Du et al., 2023; Li et al., 2023a) apply cross attention between the current utterance and its context to gain the rewriting matrix, using it for token-level actions (such as insertion, deletion, etc.). However, these methods are limited to copying tokens from the context, lack the flexibility for modifications, and also struggle with lengthy contexts. Meanwhile, many other methods (Hao et al., 2021; Xu et al., 2020; Jin et al., 2022) decompose the rewriting task into action prediction and span prediction, performing sequence tagging for action selection in each token of the current utterance and predicting candidate spans in the context. These methods do not adequately address the issue of ellipsis and coreference resolution among various similar candidates. (Inoue et al., 2022) jointly optimizes important tokens picking and generative utterances rewriting, but the design of the picker is too straightforward and still ignores the challenges.

To handle the challenges, this paper proposes a novel method called CAT (Context-Aware Tracking and Dynamic Introduction), adopting Context Tracking (CT) and Dynamic Context Introduction (DCI) for resolve ellipsis and coreference, and requisite irrelevant context filtering: 1) Context Tracking. To facilitate effective tracking and extracting the key content of complex dialogues that the current utterance depends on, we leverage a trainable tracker and propose Context Tracking to perpetually maintain a key-phrase list (kp list) for ellipsis and coreference resolution. Specifically, an empty kp list is initialized at first, as the dialogue advances, the tracker dynamically updates the kp list turn by turn. The updates include: a) add when new key phrases occur. b) delete when the topic shifts and key phrases are not dependent anymore. c) reserve when the previous key phrases may still be mentioned. 2) Dynamic Context Introduction. In order to further filter out the turns which irrelevant to the current utterance, while ensuring the contextual integrity required for rewriting, the Dynamic Context Introduction mechanism is applied. Upon the update of the key phrases list, each key phrase is tagged with the index of the most recent turn it is mentioned. Based on that, the minimal dependency context is obtained by filtering out the part preceding the furthest turn tagged in the key phrases list. Ultimately, the rewriter takes the current utterance, kp list, as well as the minimal dependency context as input to generate the final complete utterance. To fully evaluate our proposed CAT, we collect the real dialogues between users and salespersons from our platform with manual annotation and construct a new challenging benchmark INSQR. Experimental results across three datasets prove the effectiveness of our method.

Our contributions are summarized as follows:

- We propose a novel Incomplete Utterance Rewriting method, which equips a tracker to maintain a key-phrases list for ellipsis and coreference resolution, and a rewriter to generate the complete utterance.
- We design the Dynamic Context Introduction mechanism to filter out the irrelevant context in order to handle the extended multi-turn dialogue, which further enhances the performance of the Rewriter.
- Our method achieves state-of-the-art results on all three IUR benchmarks, obtaining significant improvements compared to the existing methods.

2 Preliminaries

Given a dialogue context \( C = \{C_1, C_2, ..., C_{|C|}\} \) and a following incomplete utterance \( U = \{u_1, u_2, ..., u_{|U|}\} \), where \( C_i = \{c_1, c_2, ..., c_{|C_i|}\} \) is the utterance of the \( i \)-th turn, \( u_j \) and \( c_k \) are the tokens of the utterance. The goal of IUR is to learn a mapping function,

\[
U^* = f(M, U, C),
\]

where \( U^* \) denotes a complete and standalone utterance resolving ellipsis and coreference for \( U \), \( M \) denotes the trainable model parameters. Note that the textual content newly introduced in \( U^* \) is not required to maintain strict congruence with the text appearing in \( C \), and modification is allowed to ensure semantic coherence.

3 Methodology

3.1 Overview

Figure 2 illustrates the overview of our proposed CAT, which wraps a generative rewriter \( \mathcal{R} \) with a context tracker \( \mathcal{T} \). The tracker \( \mathcal{T} \) adopts CT by maintaining a key phrase list (kp list) \( K \) turn by turn.
At the $i$-th turn, $K_i = \{k^i_1, k^i_2, \ldots, k^i_{|K_i|}\}$, where $k^i_j$ is a key phrase mentioned in the previous context or the current utterance which can be referred to by the ongoing dialogue topic and content. Meanwhile, contrary to the existing methods that input the whole context $C$ into the rewriter while ignoring the scenarios where $C$ is particularly verbose, we proposed a DCI mechanism to filter the irrelevant context base on the latest kp list. At the $i$-th turn, the input minimal dependency context is denoted as $C^\text{min}_i = \{C_m, C_{m+1}, \ldots, C_{i-1}\}$, $m \leq i - 1$, where $m$ is the furthest turn that the kp list has dependent. Eventually, the rewriter performs IUR based on the kp list of last turn $K_{i-1}$, the minimal dependency context $C^\text{min}_i$, combined with the current utterance $U_i$ to obtain the rewritten complete utterance $U^*_i$. In addition, We leverage LLM and in-context learning to generate weakly supervised labels, thereby enabling the training of the tracker and gaining the capabilities of CT by LLM distillation.

3.2 Tracker for CT

The most common way (Liu et al., 2020; Hao et al., 2021; Jin et al., 2022; Inoue et al., 2022; Li et al., 2023a) to address ellipsis and coreference resolution is to directly predict the corresponding phrases from the context $C$ for IUR, which is too straightforward to solve the challenge proposed in Section 1. Thus, we design CT that leverages a tracker $T$ to dynamically maintain a kp list $K$ turn by turn.

In particular, the kp list is first initialized as an empty list, denoted as $K_0$. In each turn of the dialogues, $K_i$ is predicted base on the last kp list $K_{i-1}$, the current utterance $U_i$, and the limited previous context $C^T_i$, denoted as

$$K_i = T(K_{i-1}, U_i, C^T_i),$$

where $C^T_i = \{C_j, C_{j+1}, \ldots, C_{i-1}\}$, $j = \max(1, i - \gamma)$, $\gamma$ is a hyper-parameter. The newly $K_i$ is edited from $K_{i-1}$ by these actions:

- **add**: insert a newly occurring key-phrase
- **reserve**: retain $k^i_{j-1} \in K_{i-1}$, while $k^i_{j-1}$ may still be referred to.
- **delete**: remove $k^i_{j-1} \in K_{i-1}$, while $k^i_{j-1}$ is not related to the current topic or content.

The update process persists across each turn. For instance, in Figure 2, "medical insurance" is added to $K_2$ mentioned in $C_2$, due to its relevance to the dialogue topic. Beginning from the $i$-th, the customer consistently consults about reimbursed "general outpatient diagnostic and treatment costs", so it is reserved persistent in the kp list. Meanwhile, as the salesperson consistently recommends different insurance products, the user’s focus shifts from one insurance product to another, such as from "Healthy Fortune: Lifetime Critical Insurance" to "Medical Excellence Outpatient Zero Deductible", and the old one that probably is not be mentioned.

![Figure 2: An overview of CAT. Initially, a tracker distilled from GPT4-turbo conducts CT to generate a key phrase list for each turn in the input context. Then, the DCI mechanism is utilized to obtain the minimal dependency context. Consequently, the rewriter takes over them combined with the incomplete utterance, and performs rewriting to output the complete utterance.](image_url)
in the after dialogue is removed while the new one is added accordingly. Ultimately, the updated kp list at the \(n\)-th turn serves as the input for rewriting the incomplete utterance in the \((n + 1)\)-th turn.

In light of the diversity of expressions for the same subject in the context, we utilize a generative way instead of span extraction. We employ T5 \(\text{(Raffel et al., 2020)}\) as the \(T\), to take over the task instruction combined with \((K_{i-1}, U_i, C^T_i)\) to create input prompts, and require the output of \(K_i\) in a predefined structured format

\[
\langle\text{bos}\rangle x_{1,1}^{kp}, x_{1,2}^{kp}, \ldots, x_{1,|k_i|}^{kp}, \langle\text{sep}\rangle, x_{2,1}^{kp}, x_{2,2}^{kp}, \ldots, x_{2,|k_j|}^{kp}, \langle\text{sep}\rangle, \ldots, x_{|k_i|}^{kp}, \langle\text{sep}\rangle, \langle\text{eos}\rangle,
\]

(3)

where \(\langle\text{bos}\rangle\) and \(\langle\text{eos}\rangle\) denotes the beginning and the end of the decoding, \(\langle\text{sep}\rangle\) is the separate token and \(x_{i,j}^{kp}\) is the \(j\)-th token of \(k_i\).

### 3.3 Rewriter Equipped with DCI

Existing work \(\text{(Inoue et al., 2022; Li et al., 2023b)}\) has proved the efficiency of generative methods for IUR, hence, we also employ T5 as the backbone for the rewriter \(R\). To handle the extended dialogues, we design a DCI mechanism to dynamically truncate the irrelevant preceding turns, instead of feeding the entire context into the model.

In brief, the DCI mechanism iteratively records index \(I = \{I_1, I_2, \ldots, I_{|K_i|}\}\) for kp list \(K_i\), which denotes the closest turn that each key phrase is mentioned. In \(i\)-th turn, the index \(I_j^i \in I_i\) of \(k_j^i \in K_i\) is obtained by the following way: 1) If \(k_j^i\) is mentioned in \(C_i\) and does not exist in \(K_{i-1}\), it means that \(k_j^i\) is a newly added key phrase, thus \(I_j^i = i\). 2) If \(k_j^i\) is not mentioned in \(C_i\) but exists in \(K_{i-1}\), it signifies that \(k_j^i\) is a reserved key phrase, hence \(I_j^i = I_{k_j^i-1}^i\), where \(k_j^i-1 \in K_{i-1}\) and \(k_j^i-1 = k_j^i\). 3) If \(k_j^i\) is mentioned in \(C_i\) and also exists in \(K_{i-1}\), this denotes that \(k_j^i\) need to update the recorded index, therefore \(I_j^i = i\). Here, to judge whether \(k_j^i\) is mentioned in \(C_i\), we perform sliding window matching on \(k_j^i\) and \(C_i\) with a window size \(|k_j^i|\), using Levenshtein distance to compute similarity, and selecting the maximum score as the similarity score \(s_j^i\). If \(s_j^i \geq \delta\), then we considered that \(k_j^i\) is mentioned in \(C_i\), where \(\delta\) is a hyper-parameter. After the iterative recording, in \((i + 1)\)-th turn, the minimal dependency context is gained as \(C_{\text{min}(2:i)}\). In Figure 2, \(K_{i\alpha}\) includes two key phrases, while the former is mentioned in the \((n - 2)\)-th turn and the latter appears in the \(n\)-th turn. So the minimal dependency context for rewriting is \(C_{n-2:n}\).

The reason we employ fuzzy matching and index recording instead of directly requiring the tracker \(T\) to predict the action of \(K_i\)’s updating is that integrating the prediction of the kp list and action would increase the decoding complexity for the rewriter. Furthermore, it would significantly diminish the accuracy of GPT4-turbo during zero-shot predictions, which is introduced in the next subsection.

During rewriting, we prompt the task instruction combined with the current incomplete utterance \(U_i\), the minimal dependency context \(C^\text{min}_i\), and the kp list \(K_{i-1}\) predicted in last turn as input. While the rewriter \(R\) directly generate the rewritten complete utterance, as follow

\[
U^*_i = R(U_i, C^\text{min}_i, K_{i-1}),
\]

(4)

where \(K_{i-1}\) provides precise candidates for ellipsis and coreference resolution, and \(C^\text{min}_i\) offers a minimal context for comprehending the topic and main content of the dialogue. Consequently, a more semantically accurate and complete \(U^*_i\) is attained.

### 3.4 Weakly Supervised Learning via LLM Distillation

LLMs recently distinguish themselves due to their potent generalization capabilities and robustness, showing great zero-shot performance on different tasks. Meanwhile, in-context learning (ICL), as a method within the prompt construction, is being widely adopted. It facilitates the model’s task understanding by providing representative input-output examples indicative of the current task, included in the prompt. To solve the problem of lacking labeled data for training \(T\) since only golden rewriting utterance can be used to ensure fairness, we leverage GPT4-turbo combined with ICL to obtain weakly supervised labels.

Specifically, for each dataset \(D\), we select \(\tau\) representative examples of dialogue context as the ICL set, denoted as \(D^\text{icl} = \{c^{\text{icl}}_1, c^{\text{icl}}_2, \ldots, c^{\text{icl}}_\tau\}\). In order to make sure that the sampled contexts are emblematic and have substantial distinctions amongst each other, we choose them relying on many factors, such as the number of dialogue turns, the topics, the complexity of the content, and so on. We manually annotate these limited examples as \(D^\text{icl} = \{ (C^\text{icl}_1, K_1), (C^\text{icl}_2, K_2), \ldots, (C^\text{icl}_\tau, K_\tau) \}\), where \(K_i = \{K^i_1, K^i_2, \ldots, K^i_{|K_i|}\}\) is a set of annotated kp list corresponding to each turn of the context \(C_i\), and \(K^j_i\) denotes the labeled kp list in \(j\)-th
turn of the context $C_i$. We prompt the $D^{icl}$ incorporate with explicit task instruction and the $C_i$ as input. The inference is formulated as

$$K^*_i = GPT(\hat{D}^{icl}, C_i),$$  \hspace{1cm} (5)$$

where $K^*_i$ is the results parsed from the output. Subsequently, we filter some error returns during requesting and obviously wrong result after parsing, and gain the weakly labeled dataset $D^* = \{(C_1, K^*_1), (C_2, K^*_2), \ldots, (C_{|D^*|}, K^*_{|D^*|})\}$ with a high quality. After that, we divide each pair of $(C_i, K^*_i) \in D^*$ into training samples $(K_{i,j}^{c_{i-1}}, U_j^{c_{i,j}}, K_{i,j}^{*_{c_{i,j}}})$ for the tracker $T$, as shown in formulation 2, where $K_{i,j}^{c_{i-1}} \in K^*_i$.

During the training process, the tracker $T$ firstly performs weakly supervised learning based on the constructed dataset, to distill the capability of GPT-4-turbo on the CT task. After that, the rewriter $R$ is trained with the $K$ of each IUR sample predicted by $T$. Here, we do not employ joint learning, since the granularity of the training samples for the tracker $T$ and rewriter $R$ is not uniform, which is also due to the consideration of efficiency.

4 Experiments

Datasets

To comprehensively evaluate our method, we used the following three datasets covering both English and Chinese: 1) **INSQR** is a challenging Chinese IUR dataset constructed by the real dialogues between customers and sales collected from Alipay\(^2\) platform, as well as the manual annotation to provide the gold utterance. Within, the longest dialogues can reach 100 turns, and the average turns is about 30, which is much longer than the existing datasets. The detailed comparison is shown in Table 1, and Table 2 shows the distribution of the number of turns.

The dataset is divided into 4733/591/593 for Train/Val/Test. 2) **CANARD** (Elgohary et al., 2019) is a representative English IUR dataset derived by QuAC (Choi et al., 2018), an open-domain conversational question answering dataset about specific Wikipedia sections, while CANARD rewrites the originally incomplete questions in QuAC. It contains 31526/3430/5571 in the Train/Val/Test set for evaluation. 3) **CQR** (Regan et al., 2019) is an English IUR dataset extended from task-oriented dialogue (Eric et al., 2017) between drivers and an in-car assistant, which is di-

\(2\)https://www.alipay.com/

Table 1: The comparison of INSQR with the existing benchmarks. Here, “avgc” and “avgq” denote the average of turns and length of context, separately. We use the three with the longest context for experiments.

<table>
<thead>
<tr>
<th>Subset</th>
<th>0-10</th>
<th>11-20</th>
<th>21-30</th>
<th>31-40</th>
<th>41-50</th>
<th>51+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>723</td>
<td>774</td>
<td>1429</td>
<td>716</td>
<td>394</td>
<td>697</td>
</tr>
<tr>
<td>Dev</td>
<td>94</td>
<td>91</td>
<td>191</td>
<td>70</td>
<td>54</td>
<td>91</td>
</tr>
<tr>
<td>Test</td>
<td>90</td>
<td>95</td>
<td>189</td>
<td>78</td>
<td>50</td>
<td>91</td>
</tr>
</tbody>
</table>

Table 2: The distribution of the number of dialogue turns in INSQR.

Table 3: The number of limited turns in INSQR.

The prompt of distillation is shown in the Github page.

---

---
### 4.1 Overall Results

Initially, we compare our proposed CAT with the existing methods. To ensure fairness, the pretrained model (BERT and T5) equipped by all the methods is set to the base version. Here, CAT denotes the method proposed in this paper including the tracker $T$ and the rewriter $R$, CAT+GPT4 signifies the direct use of results $K$ returned by GPT4-turbo, and CAT+Gold indicates the employment of golden phrases provided by the dataset, required in ellipsis and coreference resolution during rewriting. Table 3 and Table 4 show the experimental results, wherein our CAT outperforms the other methods and achieves state-of-the-art on all three datasets.

### Results on INSQR and CQR

In INSQR and CQR, the main challenges of the rewriting are ellipsis and coreference resolution, while minimally revising the rest of the original sentence. Therefore, the metrics such as BLEU and ROUGE remain at a high level. However, the Restoration score reveals the difference in the accuracy of the actual modified part, where INSQR is obviously much lower than CQR, indicating that INSQR presents more complex scenarios. Experimental results demonstrate that, in comparison to the backbone T5 model, CAT achieves a 7% and 4.1% promotion in Restoration score, respectively, which prove the accurate $K$ providing and irrelevant context filtering, can effectively improve the performance of IUR. Meanwhile, the comparison between CAT and CAT+GPT4 yields an interesting result that, the tracker $T$ surprisingly achieves a greater enhancement than GPT4-turbo in assisting the rewriter $R$. We consider the reason is that the output from GPT4-turbo including its inherent randomness and error rate on both request and return, may introduce noise and bias when it is directly used as $K$ which also serves as the foundation for DCI. In contrast, after training, $T$ predicts $K$ following its stable task understanding, hence achieving higher quality. CAT+Gold delineates the upper bound of dataset metrics. It is observable that, although the current CAT attains a notable improvement in INSQR and CQR, there remains a substantial gap to the upper bound, especially the Restoration score. It again highlights the complexity inherent in INSQR since the gap is huge.

### Results on CANARD

CANARD presents a distinct scenario compared to the above two datasets. Beyond ellipsis and coreference resolution, it also entails modifications to the key phrases themselves, and sometimes also to the rest part of the original sentences. These modifications lack a definitive uniqueness, making
it challenging for the Restoration score to evaluate the performance. The results shown in Table 4 present that the generative methods outperform the editing-based methods due to their flexibility in conforming to the various modifications present in the labels. CAT further delivers an improvement over existing methods. Within, JET also employs a picker for directly acquiring referential candidate phrases from the entire preceding context. In contrast, our proposed CT decomposes the complexity of prediction into each turn, maintaining a more precise $K$, combined with a brief context $C_{min}$ provided by DCI, thereby achieving superior performance on IUR. During the distillation, GPT4-turbo exhibits a more stable predictive performance on CANARD because all the dialogues are about Wikipedia, so the results of CAT and CAT+GPT4 are close. Here, CANARD does not provide annotations of key phrases, hence the absence of the CAT+Gold setting.

4.2 Ablation Tests

To evaluate the contributions of each component of CAT, the ablation test is constructed. Here, we consider the following settings

- w/o CT: Removing $T$ and rewriting without $K$, while $C_{min}$ is obtained by directly adopt DCI on the prediction of GPT4-turbo.
- w/o DCI: Removing the DCI while inputting the entire $C$ to $R$.

The results are shown in Table 5. On our challenge dataset INSQR, both CT and DCI demonstrate significant improvement, which shows the effectiveness of CAT to handle the scenario with confusing ellipsis and coreference resolution and extended context. In the case of CQR, which primarily focuses on ellipsis and coreference resolution with short context, CT plays a pivotal role, while the impact of DCI is less pronounced. Meanwhile, both CT and DCI also exhibit a certain improvement in CANARD while the gold rewriting labels with non-unique expressions constrains the performance of the components.

4.3 The Impact of Context Turns

For a detailed analysis of the efficacy of our proposed methods when dealing with the extended dialogues of varying turn and complexity, we partitioned the INSQR test set into four subsets based on the number of turns: 0-10, 11-30, 31-50, and 51+ (with the maximum being 100 turns). We then evaluate the score of Bleu-2 and Rouge-2 on these subsets, employing the same settings used in the ablation tests. The results are shown in Figure 3. With the increase in the number of turns, the improvement of CAT over the backbone T5 is progressively amplified, which proves the effectiveness of CAT in handling complex and extended dialogues. CT consistently delivers high contributions across all the subsets with various lengths, demonstrating that accurate $K$ can steadily improve the performance of ellipsis and coreference resolution during rewriting. Meanwhile, when the context is brief with fewer turns, particularly within 10 turns, the

<table>
<thead>
<tr>
<th>Method</th>
<th>B1</th>
<th>B2</th>
<th>R1</th>
<th>R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSQR</td>
<td>CAT</td>
<td>78.8</td>
<td>74.7</td>
<td>89.8</td>
</tr>
<tr>
<td>w/o CT</td>
<td>75.2</td>
<td>71.4</td>
<td>84.8</td>
<td>73.5</td>
</tr>
<tr>
<td>w/o DCI</td>
<td>76.8</td>
<td>72.7</td>
<td>86.4</td>
<td>76.0</td>
</tr>
<tr>
<td>T5</td>
<td>73.4</td>
<td>69.8</td>
<td>82.2</td>
<td>70.3</td>
</tr>
<tr>
<td>CQR</td>
<td>CAT</td>
<td>85.4</td>
<td>83.9</td>
<td>93.7</td>
</tr>
<tr>
<td>w/o CT</td>
<td>81.6</td>
<td>77.4</td>
<td>89.7</td>
<td>81.2</td>
</tr>
<tr>
<td>w/o DCI</td>
<td>85.4</td>
<td>83.5</td>
<td>92.7</td>
<td>87.9</td>
</tr>
<tr>
<td>T5</td>
<td>80.8</td>
<td>76.1</td>
<td>88.1</td>
<td>79.1</td>
</tr>
<tr>
<td>CANARD</td>
<td>CAT</td>
<td>79.7</td>
<td>72.7</td>
<td>84.9</td>
</tr>
<tr>
<td>w/o CT</td>
<td>77.9</td>
<td>71.5</td>
<td>82.9</td>
<td>70.6</td>
</tr>
<tr>
<td>w/o DCI</td>
<td>78.9</td>
<td>72.2</td>
<td>84.2</td>
<td>71.2</td>
</tr>
<tr>
<td>T5</td>
<td>77.1</td>
<td>70.7</td>
<td>81.9</td>
<td>69.8</td>
</tr>
</tbody>
</table>

Table 5: Overall ablation tests on three datasets.
Table 6: The results of CT. Here, P, R, and F1 denote precision score, recall score, and F1 score.

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSQR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT4-turbo</td>
<td>50.7</td>
<td>68.4</td>
<td>58.2</td>
</tr>
<tr>
<td>Tracker</td>
<td>57.2</td>
<td>64.4</td>
<td>60.6</td>
</tr>
<tr>
<td>CQR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT4-turbo</td>
<td>55.2</td>
<td>76.1</td>
<td>64.0</td>
</tr>
<tr>
<td>Tracker</td>
<td>61.6</td>
<td>70.1</td>
<td>65.6</td>
</tr>
</tbody>
</table>

enhancement brought by DCI is slight. However, after the increase in dialogues (31-50 and 50+), the demand for filtration of irrelevant context is amplified, thereby the contribution of DCI is highlighted.

4.4 Performance on CT

In addition to rewriting, we also conduct experiments to evaluate the results of CT. Herein, we compare the prediction results of $\mathcal{T}$ by the trained $\mathcal{T}$ with the zero-shot GPT-4-turbo on INSQR and CQR which both provide golden annotations for the key phrases leveraged by rewriting utterance. Note that we only evaluate the precision, recall, and F1 of the $\mathcal{T}$ on the rewriting turns due to the absence of annotations for the other turns. The results, shown in Table 6 indicate that GPT4-turbo retains the great capability to identify key phrases assisted by in-context learning, thus performing well in recall. However, it also tends to predict phrases that are extraneous to the core topic, resulting in a lower precision. In contrast to the zero-shot prediction, the trained tracker $\mathcal{T}$ demonstrates a deeper comprehension of the overall data distribution and the key phrases required to be extracted, thereby achieving a marked improvement in precision. While this comes with a little decrease in recall. Nonetheless, overall performance of $\mathcal{T}$ remains marginally superior to that of GPT-4-turbo. It also explains that, as shown in Table 3, employing the trained $\mathcal{T}$ achieves better performance than directly using the outputs of GPT-4-turbo.

5 Related Work

Early works on IUR primarily combine sequence-to-sequence based model with a copy mechanism to fetch the relevant information in the context (Su et al., 2019; Elgohary et al., 2019; Kumar and Joshi, 2017). Due to the strong representation capabilities of pre-trained models (Devlin et al., 2019), RUN (Liu et al., 2020) introduces a rewriting matrix derived from the embedding of context and utterance to adjudicate on a token-level whether insertion or replacement operations should be carried out. SRL (Xu et al., 2020) pre-identifies the candidate lexicon for the subject, predicate, and object within the context, and incorporates these as extra features into the encoder to achieve better performance. RaST (Hao et al., 2021) formulates the IUR as sequence labeling, effectuating the rewrite by predicting actions (insertion, deletion, and None) and the spans of the context separately. HCT (Jin et al., 2022) enhances the action predictor of the previous work to a rule predictor, optimizing for the generation of modified words not exactly present in the context. QUEEN (Si et al., 2022) proposed a query template that explicitly brings guided semantic structural knowledge between the incomplete utterance and the rewritten utterance. JET (Inoue et al., 2022) jointly optimizes important token picking and rewritten utterances generation (Raffel et al., 2020), but the design of the picker is too straightforward to perform accurate recall in extended dialogues. (Li et al., 2023b) employs continued pre-training on the T5 (Raffel et al., 2020) model to enhance its performance on IUR, while (Du et al., 2023) and (Li et al., 2023a) further devise fine-grained subtasks and refined model architectures based on the edit matrix. However, the existing benchmarks targeted by all the work referenced above are limited to simple ellipsis and coreference problems, while none of them pay attention to the challenges mentioned in section 1.

6 Conclusion

In this paper, we presented a novel method for incomplete utterance rewriting among extended dialogues with complex content, called CAT which is a component of a context tracker and a generative rewriter. Context Tracking is adopted by the tracker to maintain a list turn by turn, which includes the key phrase mentioned in the previous context that may still be referred to in the current utterance, as candidates to promote the performance of ellipsis and coreference resolution. Furthermore, the Dynamic Context Introduction is designed to filter the irrelevant previous turns and retain the minimal dependency context for the rewriter. Extensive experiments indicate that CAT achieves state-of-the-art results on three benchmarks compared with the existing baselines. In future work, we will expand our work on a wider range of scenarios, and also explore the optimization of the tracker.
7 Limitations

This paper proposed a novel Incomplete Utterance Rewriting method, namely CAT, which primarily focuses on the task in dialogue systems with extended multi-turn dialogue and complex content. Nevertheless, we have not combined IUR with some downstream tasks such as recommendation or question answering for further discussion. Meanwhile, though the design of Context Tracking can yield significant improvement, it necessitates real-time computation and prediction as the dialogue moves on, which results in substantial computational resources required in actual scenarios. Simultaneously, the efficacy of the tacker is heavily contingent upon the quality of weakly supervised labels obtained from the LLMs.

References


