Incomplete Utterance Rewriting by A Two-Phase Locate-and-Fill Regime

Zitong Li[1], Jiawei Li[2], Haifeng Tang[3], Kenny Q. Zhu[4]*, Ruolan Yang[5]

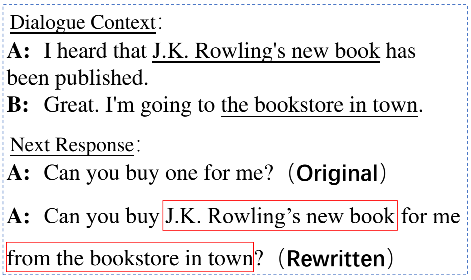
[1,2,4]Shanghai Jiao Tong University, Shanghai, China
[3]China Merchants Bank Credit Card Center, Shanghai, China
[5]University of California, San Diego, US
[1]AutSky_JadeK@sjtu.edu.cn, [2]lijiaweibecky@gmail.com
[3]thfeng@cmbchina.com, [4]kzhu@cs.sjtu.edu.cn, [5]ruy009@ucsd.edu

Abstract

Rewriting incomplete and ambiguous utterances can improve dialogue models' understanding of the context and help them generate better results. However, the existing end-to-end models will have the problem of too large search space, resulting in poor quality of rewriting results. We propose a 2-phase rewriting framework which first predicts the empty slots in the utterance that need to be completed, and then generate the part to be filled into each positions. Our framework is simple to implement, fast to run, and achieves the state-of-the-art results on several public rewriting datasets.

1 Introduction

In multi-turn dialogues, speakers naturally tend to make heavy use of references or omit complex discourses to save the efforts. Thus natural language understanding models usually need the dialogue history to understand the true meaning of the current utterance. The existence of such incomplete utterances increases the difficulty of modeling dialogues.



Dialogue Context:
A: I heard that J.K. Rowling's new book has been published.
B: Great. I'm going to the bookstore in town.

Next Response:
A: Can you buy one for me? (**Original**)
A: Can you buy J.K. Rowling's new book for me from the bookstore in town? (**Rewritten**)

Figure 1: An example of utterance rewriting. The phrase in the first red box is coreference, and the second is ellipsis.

The sources of incompleteness of an utterance can be divided into two categories: *coreference* and *ellipsis*. The task for solving these two kinds of incompleteness is called Incomplete Utterance Rewriting (IUR). As shown in Figure 1, the third utterance of this multi-turn dialogue is incomplete. If this utterance is taken out alone without a context, we will not be able to understand what "one" means and where to buy it. The fourth utterance is a rewriting of the third one. We can see that "one" in the third utterance is replaced by "J.K. Rowling's new book". In addition, the place adverbial "from the book store in town" is inserted after "for me". In today's industry strength dialogue systems and applications, due to stringent requirements on running time and maintenance cost, single-turn models are much more preferred than multi-turn models. If an incomplete single-turn utterance can be completed, it will be more understandable without the context, and the cost of downstream NLP tasks, such as intention extraction and response generation, will be reduced.

Figure 1 shows that that all the words added in the rewritten utterance except "from" come from the context. Inspired by this, many early rewriting works used pointer networks (Vinyals et al., 2015) or sequence to sequence models with copy mechanism (Gu et al., 2016; See et al., 2017) to directly copy parts from the context into the target utterance. More recently, pre-trained language models such as T5 (Raffel et al., 2020) succeeds in many NLP tasks, and it appears that T5 is a plausible choice for utterance rewriting as well. However, IUR task is different from other generation tasks in that new parts typically only need to be added in one or two specific locations in the original utterance. That is, the changes to the utterance are localized. For example, a typical operation is adding modifiers before or after a noun. On the contrary, end-to-end text generation models such as T5 may not preserve the syntactic structure of the input, which may cause the loss of important information and the introduction of wrong information into the output, which is illustrated as below (Two examples are generated by T5.).

[*] The corresponding author.

- Can you buy **J.K. Rowling's new book**? (Losing original structure)

- Can you **publish** new book for me ? (Introducing wrong information)

Another problem of the end-to-end pre-trained models, which generate the rewritten utterances from scratch, is that they generally incur a large search space and are therefore not only imprecise but also inefficient. In order to solve the large search space issue, Hao et al. (2021a) treated utterance rewriting as a sequence tagging task. For each input word, they predict whether it should be deleted and the span that needs to be replaced with. Liu et al. (2020) formulated IUR as a syntactic segmentation task. They predict segmentation operations required on the utterance to be rewritten. However, they still did not take the important step of predicting the site of rewrite, particularly the position within the syntactic structure of the input utterance. If the model can learn the syntactic structure information in the target sentence, it can predict which part of the sentence needs to be modified, i.e., which words need to be replaced and where new words need to be inserted. After that, the model only needs to fill in these predicted positions. These two tasks are relatively simple to perform, and they collectively avoid the above problems. Our approach is based on the above intuition.

In order to effectively utilize the syntactic structure of the sentence to be rewritten, we divide the IUR task into two phases. The first phase is to predict which positions in the utterance need to be rewritten (including coreference and ellipsis). The second phase is to fill in the predicted positions. In the first phase, we use the sequence annotation method to predict the locations of coreference and ellipsis in the utterance. In the second phase, we take the utterances with blanks as input and directly predict the words required for the blank position. By seperating the original rewriting task into two relatively simple phases, our results show that our model performs the best among recent state-of-the-art rewriting models [1].

Our main contributions are as follows.

- A two-phase framework for solving incomplete utterance rewriting task is proposed. It can complete the Incomplete Utterance Rewriting (IUR) task. (Section 2)

- An algorithm for aligning the two sentences before and after rewriting based on the longest common subsequences (LCS) algorithm. We succinctly and efficiently generated two kinds of data which can be used for predicting the positions to be rewritten (the first phase) and filling the blanks (the second phase) respectively. (Section 2.1.2)

- We have carried out experiments on 5 datasets, and the experimental results show that our two-phase framework achieves state-of-the-art results. (Section 3)

2 Approach

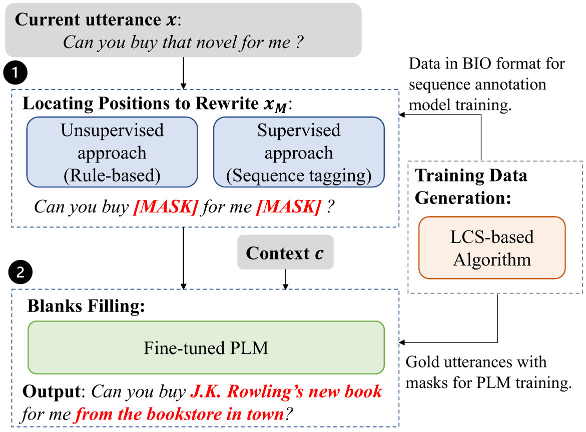

Current utterance x:
Can you buy that novel for me ?
1
Locating Positions to Rewrite x_M:
Unsupervised approach (Rule-based)
Supervised approach (Sequence tagging)
Can you buy ***[MASK]*** *for me* ***[MASK]*** *?*
Data in BIO format for sequence annotation model training.
Training Data Generation:
LCS-based Algorithm
Context c
2
Blanks Filling:
Fine-tuned PLM
Gold utterances with masks for PLM training.
Output: *Can you buy* ***J.K. Rowling's new book*** *for me* ***from the bookstore in town****?*

Figure 2: Our 2-phase rewrite framework.

Our framework is divided into two phases: **Locating positions to rewrite** and **Filling the blanks**. Figure 2 is a brief schematic of the framework. Phase 1 can be done either by heuristic rules or by supervision. Phase 2 can be done with a seq2seq text generation model. We give the details of these phases next.

2.1 Locating Positions to Rewrite

We designed an unsupervised and a supervised method to locate positions to rewrite. The two methods are described below.

2.1.1 Unsupervised Rule-based Method

We first implement a rule-based method for the first phase of our problem, aiming at predicting the blanks automatically. We looked through thousands of complete utterance examples in Elgohary et al. (2019). Based on our observations and experience, we define six rules for generating two kinds

[1] Complete code is available at `https://github.com/AutSky-JadeK/Locate-and-Fill`.

of blanks which are used for resolving coreference and ellpisis in the second phase. The rules for generating blanks are summarized and explained below:

Personal Pronouns: We replace all the personal pronouns (except the first- and second-person pronouns) and their corresponding possessive pronouns with [MASK_r]. This indicates that we will replace these pronouns with some specific noun phrases at second phase.

Interrogatives: We insert [MASK_i] after the interrogative if the whole utterance only contains interrogatives such as what, how, why, when and so forth. [MASK_i] indicates that some additional text span shall be inserted at this location.

That, This: The use of word like "this", "that", "these" and "those" are commonly used in colloquial language, which becomes a source of ambiguity. Therefore, we deal with the use of these pronouns in following ways:

- Not followed by a noun phrase: In this case, we simply replace the word by [MASK_r].
- Otherwise: We will insert [MASK_i] after the noun phrase.

The+Noun Phrase: We will insert [MASK_i] after the noun phrase.

Other, Another, Else: If the utterance contains these words, it usually indicates that there are people/things additional to what have been mentioned before. Hence, we add a [MASK_i] at the head of the sentence.

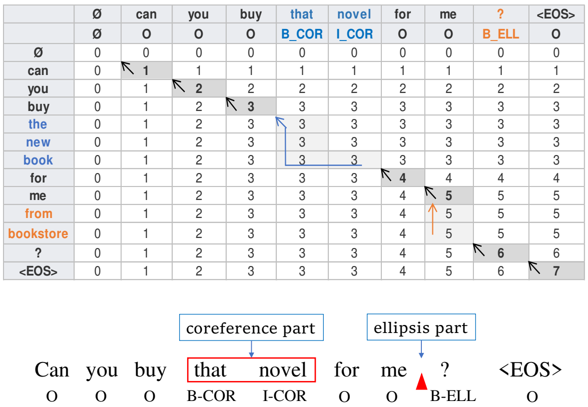
Before, After: We insert [MASK_i] after the sentence ended with "before" or "after", which is considered as an incompletion.

2.1.2 Supervised LCS-based Method

We also design an algorithm based on the Longest Common Subsequence (LCS) algorithm . The sentence to be rewritten **X** and after rewriting **Y** are aligned via a sequence labeling model. To obtain the common subsequence, LCS algorithm returns a matrix **M** which stores the LCS sequence for each step of the calculation. The value of $M_{i,j}$ indicates the actual LCS length of sequences $X[0, i]$ and $Y[0, j]$ [2]. When we trace back from the max value at the corner, the decreases of length show that the sentences have a common token.

Coreference and ellipsis towards original sentence are extracted through LCS trace back algorithm, which is further labeled as COR and ELL respectively. Given the tokenized original sentence **X** and ground truth **Y** as shown in Figure 3, the rules for labeling are specified as follows:

- The labeling is proceeded from the bottom right to the top left corner of a LCS matrix. If the current tokens in X_i and Y_j are equal, X_i matches part of the LCS and is labeled as O, then we go both up and left (shown in black). If not, we go up or left, depending on which cell has a higher number or lower index j, until we find next matched $X_{i'}$ that satisfies $X_{i'} = Y_{j'}$.
- If traversed path from previously matched token pair to newly match pair is a straight up arrow, it indicates that token(s) in interval $(Y_{j'}, Y_j)$ [3] is (are) inserted at corresponding position i' in **X** to complete the original sentence. In this case, token $X_{i'}$ is labeled as ELL(shown in orange).
- If two matched pairs in the LCS matrix are joined by paths with corners, interval $(X_{i'}, X_i)$ is replaced by $(Y_{j'}, Y_j)$ during rewriting. As a result, coreferenced words are labeled as COR(shown in blue).



	Ø	can	you	buy	that	novel	for	me	?	<EOS>
	Ø	O	O	O	B_COR	I_COR	O	O	B_ELL	O
Ø	0	0	0	0	0	0	0	0	0	0
can	0	1	1	1	1	1	1	1	1	1
you	0	1	2	2	2	2	2	2	2	2
buy	0	1	2	3	3	3	3	3	3	3
the	0	1	2	3	3	3	3	3	3	3
new	0	1	2	3	3	3	3	3	3	3
book	0	1	2	3	3	3	3	3	3	3
for	0	1	2	3	3	3	4	4	4	4
me	0	1	2	3	3	3	4	5	5	5
from	0	1	2	3	3	3	4	5	5	5
bookstore	0	1	2	3	3	3	4	5	5	5
?	0	1	2	3	3	3	4	5	6	6
<EOS>	0	1	2	3	3	3	4	5	6	7

coreference part
ellipsis part
Can you buy that novel for me ? <EOS>
O O O B-COR I-COR O O B-ELL O

Figure 3: Example of generating sequence labeling data (based on LCS).

Then, we input the pre-processed training data into the BERT-CRF (Souza et al., 2019) model, which is considered as a sequence annotation task. Using the method described in Section 2.1.2, we obtained the locations of coreferences and ellipses

[2] https://en.wikipedia.org/wiki/Longest_common_subsequence_problem

[3] (a, b) means an open interval excl. the endpoints a and b.

of each utterance waiting to be rewritten. As shown in Figure 3, we use the BIO format to annotate the sequence. The starting position of the coreference is marked as B-COR (Begin-Coreference), while other positions of the coreference are marked as I-CORs (Inside-Coreference). Ellipsis only appears in the middle of two tokens, so we mark the position of the latter token as B-ELL (Begin-Ellipse), which means that there should be missing words between this token and the previous token, and the subsequent model is required to fill in it.

2.2 Blanks Filling

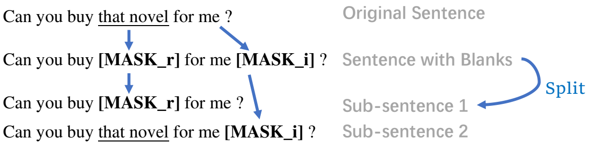

Can you buy that novel for me ?
Original Sentence
Can you buy [MASK_r] for me [MASK_i] ?
Sentence with Blanks
Split
Can you buy [MASK_r] for me ?
Sub-sentence 1
Can you buy that novel for me [MASK_i] ?
Sub-sentence 2

Figure 4: Split the sentence according to the number of blanks in the utterance.

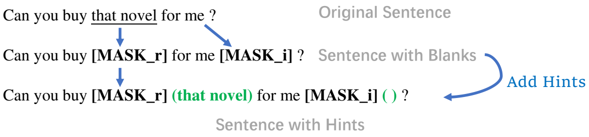

Can you buy that novel for me ?
Original Sentence
Can you buy [MASK_r] for me [MASK_i] ?
Sentence with Blanks
Add Hints
Can you buy [MASK_r] (that novel) for me [MASK_i] ( ) ?
Sentence with Hints

Figure 5: Add hints to blanks.

We use T5-small (Raffel et al., 2020) and bart-base (Lewis et al., 2020) as pre-trained language model (PLM) in phase 2. In this section, we will take T5 as an example to illustrate the process of blanks filling.

We use the T5 model to fill in blanks with two optimizations: adding hints and splitting the current utterance into sub-sentences. The latter can ensure that there is only one blank in the sentence to be filled in the T5 model. The two optimizations are shown in Figure 4 and Figure 5. We transfer the data of each multi-turn dialogue into the format shown in Figure 6, and fine-tune the T5 model.

Input: I heard that J.K. Rowling's new book has been published. *[SEP]* Great. I'm going to the bookstore in town. *[SEP]* Can you buy **<extra_id_0>** **(that book)** for me?
Output: J.K. Rowling’s new book

Input: I heard that J.K. Rowling's new book has been published. *[SEP]* Great. I'm going to the bookstore in town. *[SEP]* Can you buy that book for me **<extra_id_0>** ( ) ?
Output: from the bookstore in town

Figure 6: Format of fine-tune data of T5.

After fine-tuning, we take the predicted results of BERT-CRF model in Section 2.1.2 as input to get the final blank filled results of T5 model. Finally, the outputs of T5 model are filled back into the blanks of the original sentence to get the rewritten utterance. The same is for the rule-based method. The blank prediction obtained from it is directly input into the same T5 model (the two optimization methods described in Figure 4 and Figure 5 will also be used) to obtain the output of T5.

3 Experiment

In this section, we will introduce our experiment setup and results.

	MuDoCo	CQR	REWRITE	RES	CANARD
Train	2.39k	0.52k	16.00k	193.77k	16.88k
Dev	0.29k	0.06k	2.00k	5.10k	1.79k
Test	0.30k	0.06k	2.00k	5.10k	2.96k
Ave Len	73.43	143.70	36.85	68.38	429.77
% RW	26.68	98.38	99.98	60.00	92.91

Table 1: Descriptions of the datasets. “Ave Len” means the average length of context. “% RW” denotes the percentage of samples whose current utterance is actually rewritten.

3.1 Datasets

We tested the baseline and our framework on 3 public datasets in English and 2 in Chinese. The statistics are shown in Table 1. The examples are shown in Appendix.

MuDoCo (Martin et al., 2020) has a lower rewriting rate, which makes the rule-based method less accurate in predicting the locations to be rewritten. **CQR** (Regan et al., 2019) contains imperative dialogues in life (between people or between people and intelligent agents). The sentence patterns are relatively simple, fixed and easy to understand. **REWRITE** (Su et al., 2019a) is a Chinese dataset, each dialogue of which contains 3 turns. It is collected from several popular Chinese social media platforms. The task is to complete the last turn. **RES** (Restoration-200k) (Pan et al., 2019a) is a large-scale Chinese dataset in which 200K multi-turn conversations are collected and manually labeled with the explicit relations between an utterance and its context. Each dialogue is longer than REWRITE. **CANARD** (Elgohary et al., 2019) contains a series of English dialogues about a certain topic or person organized in the form of QA. It has the largest size and the longest context length. The sentence pattern in CANARD is complex, the understanding is difficult, and the rewriting degree is high.

3.2 Baselines

We choose the following strong baselines to compete with our framework.

T5-small model and **T5-base model** [4] (Raffel et al., 2020). We directly take the context and the current utterance as inputs, use the training set to fine-tune the T5 model, and test its end-to-end output on the test set as the result of rewriting the utterance.

BART-base model (Lewis et al., 2020). This is another pre-trained model we used. Its size is close to T5-small. Our model is tested based on these 2 PLMs.

Rewritten U-shaped Network (RUN) (Liu et al., 2020). In this work, the authors regard the incomplete utterance rewriting task as a dialogue editing task, and propose a new model using syntactic segmentation to solve this task.

Hierarchical Context Tagging (HCT)(Lisa et al., 2022). A method based on sequence tagging is proposed to solve the robustness problem in rewriting task.

Rewriting as Sequence Tagging (RAST)(Hao et al., 2021b). The authors proposed a novel tagging-based approach that results in a significantly smaller search space than the existing methods on the incomplete utterance rewriting task.

3.3 Evaluation Metrics

We use the **BLEU**$_n$ score (Papineni et al., 2002) to measure the similarity between the generated rewritten utterance and the ground truth. Low order n-gram **BLEU**$_n$ score can measure precision, while high-order n-gram can measure the fluency of the sentence. We also use the **ROUGE**$_n$ score (Lin, 2004) to measure recall of rewritten utterance. **Rewriting F-score**$_n$ (Pan et al., 2019b) is used to examine the words newly added to the current sentence. We calculte Rewriting F-score by comparing words added by the rewriting model with added words in ground truth. It is a widely accepted metric that can better measure the quality of rewriting. In addition to the automatic evaluation method, we also asked human annotators to conduct comparative tests on the rewriting results.

3.4 Implementation Details

All of the models are running and evaluated on 2 Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz with 4 NVIDIA GeForce RTX 2080 GPU and a 128GB RAM. Due to the memory constraints of our experimental environment, we adopt T5-small model in the second phase of our framework, and fine-tune it for 20 epochs. All experiments are repeated for 3 times and averaged.

3.5 Main Results

In the following section, "Ours-T5" represents our model based on T5-small in phase 2. "Ours-BART" is our framework based on BART-base in phase 2. "Ours-rule" is a variant of our method which uses the rule-based method in Section 2.1.1 to generate blanks in phase 1 and T5-small in phase 2. "Gold-T5" is the result of directly inputting the sentence with the correct blanks into the T5-small model in phase 2. "Gold-BART" is directly inputting the sentence with the correct blanks into the BART-base model in phase 2.

Table 2 shows the results of our framework and baselines on CQR and MuDoCo. Compared with CANARD, the two datasets are smaller in size and simpler in sentence structure. Our approach is significantly better than all baselines on all metrics. For **Rewriting F-score**, our method is 6.37 and 6.63 percentage points higher than the suboptimal end-to-end T5-small model, respectively. This metric strongly shows that our method can introduce more new words provided in the ground truth (compared with the original sentence). Relatively larger advantages of our model compared with T5-small in **BLEU** and **ROUGE** show that our method based on blank prediction and filling can retain the structure of the original sentence to the greatest extent, so as to retain more correct same information when calculating these two metrics and comparing the two sequences. However, end-to-end T5 model generates the whole rewriting utterance directly, which may lose some information from the original sentence.

The last part of Table 2 shows the results of our framework and baselines on CANARD. Among the three datasets we used, samples in CANARD are the most difficult and the most complex. Our model is superior to other baseline methods in all the experimental metrics. Especially in BLEU score, our method is significantly better than all baselines. As for Rewriting F-score and ROUGE, we found that

[4] A fine-tuned version of T5-base is used: `https://huggingface.co/castorini/t5-base-canard`.

Datasets	CQR			MuDoCo			CANARD		
Methods	$\mathbf{F}_{1/2}$	$\mathbf{BLEU}_{1/2}$	$\mathbf{ROUGE}_{1/2/L}$	$\mathbf{F}_{1/2}$	$\mathbf{BLEU}_{1/2}$	$\mathbf{ROUGE}_{1/2/L}$	$\mathbf{F}_{1/2}$	$\mathbf{BLEU}_{1/2}$	$\mathbf{ROUGE}_{1/2/L}$
HCT	58.6/32.3	64.2/52.9	67.8/47.3/65.6	56.1/49.2	93.0/90.7	94.9/87.8/94.9	33.9/28.4	67.9/61.7	80.1/66.5/79.5
RUN	54.0/29.5	63.1/51.9	67.3/45.1/64.3	44.8/32.0	93.0/90.2	94.4/85.4/94.3	43.8/30.5	70.1/62.2	80.5/62.9/79.0
RAST	60.9/33.7	65.4/53.8	69.0/50.6/67.7	58.9/50.7	92.4/89.9	94.0/84.7/93.8	44.8/30.8	70.5/62.9	80.6/63.8/79.4
T5-small	80.8/72.3	62.3/59.9	84.3/76.8/83.0	62.4/56.7	87.5/79.4	95.0/88.4/94.9	51.5/40.4	70.4/64.1	80.2/66.6/78.1
BART-base	79.4/71.7	61.5/57.4	82.0/74.4/81.9	60.8/54.9	85.6/78.4	93.9/87.3/93.7	52.3/41.2	68.8/62.6	78.9/65.5/77.0
Ours-rule	65.0/57.7	69.5/65.2	72.3/60.3/69.7	60.4/47.4	83.0/78.3	92.3/80.6/92.0	51.8/40.5	70.9/64.6	80.8/67.0/79.0
Ours-T5	**87.5/80.3**	**88.6/85.8**	**91.2/83.9/89.9**	**68.8/62.5**	**95.6/94.1**	**96.1/89.5/96.1**	**53.4/41.4**	**77.5/70.1**	**82.8/68.3/81.1**
Ours-BART	86.1/78.3	86.9/83.9	90.1/81.8/88.4	66.6/61.5	94.3/92.8	94.3/87.8/95.0	53.1/40.9	76.5/69.6	82.0/67.4/80.0
Gold-T5	89.3/82.9	91.3/89.0	93.6/88.0/93.2	75.9/69.7	97.4/96.3	97.8/92.2/97.8	58.2/47.9	80.1/71.3	86.2/70.0/83.1
Gold-BART	89.0/82.4	90.7/88.2	92.3/87.1/92.4	72.6/67.0	95.6/94.5	95.5/90.1/94.8	57.7/47.5	79.9/71.0	85.6/69.4/82.2

Table 2: Results on English datasets.

Datasets	REWRITE			RES		
Methods	$\mathbf{F}_{1/2}$	$\mathbf{BLEU}_{1/2}$	$\mathbf{ROUGE}_{1/2/L}$	$\mathbf{F}_{1/2}$	$\mathbf{BLEU}_{1/2}$	$\mathbf{ROUGE}_{1/2/L}$
HCT	79.3/74.2	92.7/90.2	94.4/89.3/93.5	73.2/67.1	92.1/91.7	93.4/88.8/92.8
RUN	80.5/75.0	93.5/90.9	95.8/90.3/91.3	72.9/66.9	92.0/89.1	92.1/85.4/89.5
RAST	77.8/72.5	90.5/88.3	94.7/88.9/92.9	71.8/65.3	89.7/88.8	91.1/84.2/87.8
BART-base	81.2/76.0	93.9/90.8	95.2/91.8/92.4	75.0/69.7	92.8/88.7	92.6/88.2/90.3
Ours-rule	79.1/73.8	90.2/87.8	93.3/90.6/91.4	72.3/65.8	90.5/86.3	90.4/86.1/88.5
Ours-BART	**83.4/79.1**	**94.7/92.8**	**96.0/92.2/93.7**	**76.4/70.5**	**94.3/91.9**	**95.3/89.8/91.4**
Gold-BART	85.6/80.9	95.6/93.3	94.7/92.6/92.8	80.8/73.2	95.0/91.2	95.8/91.4/92.1

Table 3: Results on Chinese datasets.

the performance of end-to-end T5 model is close to our method. This is because the generative T5 model is very powerful and can generate fluent sentences. However, our 2-phase framework can better predict which positions in the current sentence should be rewritten, which can not be achieved by the end-to-end model. In the following analysis, we will further analyze this point.

An important reason why our framework is better than baselines on CQR and MuDoCo is that CQR mainly contains dialogues that users are asking agents for help. The positions and forms of words that can be added are relatively fixed, such as adding place adverbials. Samples in MuDoCo are basically imperative dialogues in daily life. It also has the same feature, which makes our model easier to learn. The results in Section 3.7 can also illustrate this point. The accuracy of the first phase of our framework is higher on CQR and MuDoCo.

Table 3 shows the results of our framework and baselines on Chinese datasets REWRITE and RES. Due to the better performance of BART in Chinese texts, our model is mainly tested based on BART-base rather than T5-small in these two datasets. These two PLMs have similar sizes. HCT, RUN and RAST perform well on these two datasets. Because these two datasets have few turns and simple contents, they have been studied a lot in previous work. However, their performance is not as good as that of BART-base. This shows the great potential of using PLMs directly in rewriting tasks. Compared with BART-base, our model has improved in BLEU score and ROUGE score. This shows that our method is also effective in Chinese. And when different PLMs are used as frameworks, the results can be improved.

Context	**A:** yogi berra **B:**major leagues **A:**what team signed him ? **B:**berra was called up to the yankees and played his first game on september 22 , 1946 ;
Current	**A:** how long was he there ?
Gold	**A:** how long was yogi berra with the yankees ?
Ours-sup	**A:** how long was yogi berra at the yankees ?
T5-small	**A:** how long was yogi berra there ?

Table 4: A typical example extracted from the prediction results on CANARD.

In the Table 4, our model is compared with T5 model for end-to-end prediction. It can be observed that the word “there” is not considered to be replaced by the end-to-end model, which is due to the fact that the position to be rewritten is not obviously predicted. Our two-phase framework can make up for this. The sequence annotation model indicates that “there” is a part that needs to be replaced, so the T5 in the second phase can be predicted correctly. This is our advantage over the end-to-end model. More case studies are shown in appendix.

3.6 Human Evaluation

	Win	Tie	Loss
Ours v.s. HCT	0.66	0.10	0.24
Ours v.s. RUN	0.50	0.16	0.34
Ours v.s. Rule	0.70	0.10	0.20
Ours v.s. T5-small	0.46	0.16	0.38

Table 5: Human evaluation on CANARD. "Rule" means our rule based method conacted with T5-small of 2nd phase, introduced in Section 2.1.1.

Table 5 shows the results of human evaluation on CANARD. For each pair of competing models, 50 pairs of rewriting results were randomly sampled from the testset for comparative testing. A total of 200 questions were randomly assigned to 5 human volunteers on average. Each person needs to choose the better one from the prediction results of the two models. As can be seen from the table, our method is significantly stronger than RUN, HCT, and the rule-based method in Section 2.1.1. When compared with the end-to-end T5-small model, our advantage is relatively small. After observing the feedback of human annotators, we find that the end-to-end model has the advantage of direct generation and can generate more complete and fluent sentences. Our method only generates the words needed in blank, which lacks a certain degree of sentence fluency. However, our 2-phase framework can accurately predict the positions that need to be rewritten in the current sentence, which is beyond the reach of the end-to-end model (see appendix for specific analysis). Taken together, our method should be even better.

3.7 Ablation Tests

Variant	F$_1$	F$_2$	B$_1$	B$_2$	R$_1$	R$_2$	R$_L$
Ours-T5	**53.4**	**41.4**	**77.5**	**70.1**	**82.8**	**68.3**	**81.1**
w/o LCS	52.7	40.7	76.2	68.6	82.5	67.7	**81.1**
w/o split	50.4	39.2	76.6	67.5	82.5	65.2	78.8
w/o hint	52.1	40.2	76.7	69.4	82.4	67.7	80.8

Table 6: Ablation test on CANARD. "w/o LCS" means replace LCS algorithm with a greedy algorithm. "w/o split" and "w/o hint" respectively represent removing the 2 kinds of optimizations in Section 2.2.

Table 6 shows the results of end-to-end ablation test on CANARD. We can see that by replacing LCS algorithm with greedy algorithm, the experimental results have decreased to a certain extent, which shows the effectiveness of LCS algorithm. On the other hand, due to the diversity of experimental data, the matching algorithm can only approach the correct results, and can not guarantee the complete correctness. Greedy algorithm is also a substitute. Our greedy algorithm is described as follows.

We use 2 pointers to traverse the current utterance and ground truth utterance. The pointers point to the current word in each of the two utterances. If they cannot be matched, the pointer of the ground truth will advance to the next matching position and stop, and the scanned span will be marked as an "ellipsis". If no match can be made until the end, the pointer of the current occurrence moves forward one bit and adds the previous position to the span of "coreference".

If we remove the two optimizations of splitting sentences according to the number of blanks and adding hint from our framework, there will be more obvious decline. The reason is that splitting sentences can keep more syntactic information in sentences, and multiple blanks will make sentences look "full of loopholes". Adding hint will prompt the original words in the language model in phase 2, so as to provide more information. For example, if our hint is "he", the model will not tend to fill in a female name or other things here.

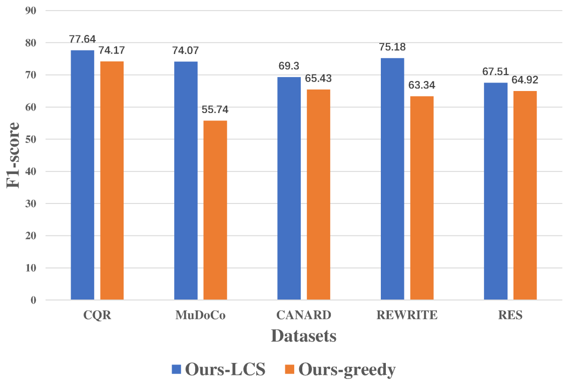


Figure 7: Different methods' results of predicting locations to be rewritten in the phase 1.

Table 7 shows the F1-score of our LCS based algorithm and greedy based algorithm in predicting the locations that need to be rewritten (that is, the first phase in the 2-phase framework). They are trained and tested on the sequence annotation data generated by their own methods. We can see that the algorithm based on LCS has better effect.

3.8 Time Cost Evaluation

Table 7 shows the results of training and predicting time on CANARD. In Section 3.5, we found that

Model	Phase 1	Phase 2	Total Time
Ours-T5	9m17s	4h31m20s	4h40m37s
T5-small	-	4h2m54s	4h2m54s

(a) Training time.

Model	Phase 1	Phase 2	Total Time	Ave Time
Ours-T5	24s	18m10s	18m34s	0.20s
T5-small	-	24m32s	24m32s	0.26s

(b) Inference Time.

Table 7: Time cost of our method and end-to-end T5-small model on CANARD. "Total Time" is the total training time spent on all samples in the test set. "Ave Time" is the average inference time of all samples in the test set.

our model has the least advantage over the end-to-end T5-small model. Therefore, in this section, we compare their time consumption. In Table 7a, under the same configuration, we found that our method would take more time to fine-tune. This is understandable because although there are only 5571 samples in the testset of canard dataset, we will segment sentences according to the number of blanks. Even if there are sentences without any blanks, this optimization also leads to an increase in the number of samples to 6569. Interestingly, in the inference time, Table 7b shows that our model takes less time. This may be because our model does not need to generate a whole sentence, but only needs to fill in the blank, which is much shorter than a complete utterance. Due to the short time of BERT-CRF, our method only takes 11.9% more time than the end-to-end T5 model, and the overall size of the model is almost the same as other training requirements. Therefore, we believe that even a small increase in results can illustrate the effectiveness of our method.

3.9 Comparison with ChatGPT

In this section, we will present the results of comparison with ChatGPT [5]. Dialogue systems are useful in many tasks and scenarios. Rewriting utterances is particularly useful when a light-weight dialogue model which only takes the last utterance as input is desirable. This is exactly where very large models such as ChatGPT cannot help, not to mention the various woes of current ChatGPT such as the cost of deployment, slow inference speed, and privacy issues. Therefore, we believe that it is not fair to compare ChatGPT with the kind of rewriting technology that we are advocating in this paper, and the latter still has its merits.

> Please complete the following incomplete sentence completion task. Given the context of the conversation and incomplete sentences to be rewritten, you need to complete the sentences to be rewritten so that they can be understood out of context. Please do not change the words in the sentence to be rewritten or the structure of the sentence unless necessary. Do not use information that goes beyond the context. Your answer should be at most 10 words more than the sentence to be rewritten.
>
> give an example:
>
> **context:**
> anna politkovskaya
> the murder remains unsolved , 2016
>
> **sentence to be rewritten:**
> did they have any clues ?
>
> **answer:**
> did investigators have any clues in the unresolved murder of anna politkovskaya ?
>
> If you understand, I will give you some tasks.

Figure 8: A prompt designed to allow ChatGPT to do rewriting task.

The scale of ChatGPT or is at least 3 orders of magnitude larger than the models we use in this paper, which means this is not a fair comparison. Nevertheless, we still conducted the following supplementary experiments on ChatGPT. The prompt we used is shown in Table 8.

Methods	F$_{1/2}$	BLEU$_{1/2}$	ROUGE$_{1/2/L}$
Ours-T5	**46.6/33.3**	**63.5/53.9**	**67.6/49.4/64.2**
ChatGPT	41.8/23.4	45.0/30.1	52.2/23.3/46.3

Table 8: Experimental results on 30 cases of CANARD.

The experimental results on 30 cases of CANARD is shown in Table 8. Some examples of the results are shown in Table 9. After repeated tries and with the best prompt we can find, ChatGPT is still worse than our method in terms of automatic evaluation metrics. However, by human evaluation, testers think that the rewriting results of ChatGPT are of higher quality (more fluent). This is no surprise given the tremendous parameter space of ChatGPT.

4 Related Work

Early work on rewriting often considers the problem as a standard text generation task, using pointer networks or sequence-to-sequence models with a copy mechanism (Su et al., 2019b; Elgohary et al.,

[5] https://chat.openai.com/

Ours-T5	did fsb get into trouble for the attack against the account annapolitovskaya@us provider1 ? why did superstar billy graham return to the wwwf ?
ChatGPT	Did the perpetrators face consequences for the attack on Anna Politkovskaya's email? What was the reason for Superstar Billy Graham's return to WWWF?

Table 9: Examples of ChatGPT and ours on CANARD.

2019; Quan et al., 2019) to fetch the relevant information in the context (Gu et al., 2016). Later, pre-trained models like T5 (Raffel et al., 2020) are fine-tuned with conversational query reformulation dataset to generate the rewritten utterance directly. Inoue et al. (2022) uses Picker which identifies the omitted tokens to optimize T5.In general, these generative approaches ignore the characteristic of IUR problem: rewritten utterances often share the same syntactic structure as the original incomplete utterances.

Given that coreference is a major source of incompleteness of an utterance, another common thought is to utilize a coreference resolution or corresponding feartures. Tseng et al. (2021) proposed a model which jointly learns coreference resolution and query rewrite with the GPT-2 architecture (Radford et al., 2019). By first predicting coreference links between the query and context, the performance of rewriting has improved while the incompleteness is induced by coreference. However, this does not work for utterances with ellipisis. Besides, the performance of the rewriting model is limited by the coreference resolution model.

Recently, some of the work on incomplete utterance rewriting focuses on the "actions" we take to change the original incomplete utterance into a self-contained utterance (target utterance). Hao et al. (2021a) solves this problem with a sequence-tagging model. For each word in the input utterance, the model will predict whether to delete it or not, meanwhile, the span of words which need to be inserted before the current word will be chosen from the context. Liu et al. (2020) formulated the problem as a syntactic segmentation task by predicting segmentation operations for the rewritten utterance. Zhang et al. (2022) extracts the coreference and omission relationship directly from the self-attention weight matrix of the transformer instead of word embeddings. Compared with these methods, our framework separates the two phases more thoroughly of predicting the rewriting position and filling in the blanks, and meanwhile, reduces the difficulty of the two phases with the divide and conquer method.

5 Conclusion

In this work, we present a new 2-phase framework which includes locating positions to rewrite and filling the blanks for solving Incomplete Utterance Rewriting (IUR) task. We also propose an LCS based method to align the original incomplete sentence with the ground truth utterance to obtain the positions of coreference and ellipsis. Results show that our model performs the best in several metrics. We also recognize two directions for further research. First, as the performance of our 2-phase framework is often limited by the first phase, we will try to improve the accuracy of locating rewriting positions. Second, it will be useful to study the best way for applying our rewriting model to other downstream NLP tasks.

6 Limitations

Our framework is a two-phase process, which has its inherent defects, that is, the results of the second phase depend on the results of the phase 1. Because the sequence annotation algorithm in the first phase cannot achieve 100% accuracy, it will predict the wrong position that should be rewritten when the second phase is followed, which will further lead to the error of the final result.

On the other hand, T5 model is only used to predict the words that should be filled in blank, rather than generate the whole sentence, which may lead to the decline of the overall fluency of the sentence.

Acknowledgments

This work was generously supported by the CMB Credit Card Center & SJTU joint research grant, and Meituan-SJTU joint research grant.

References

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

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

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021b. RAST: domain-robust dialogue rewriting as sequence tagging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4913–4924. Association for Computational Linguistics.

Shumpei Inoue, Tsungwei Liu, Nguyen Hong Son, and Minh-Tien Nguyen. 2022. Enhance incomplete utterance restoration by joint learning token extraction and text generation. *arXiv preprint arXiv:2204.03958*.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

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

Jin Lisa, Song Linfeng, Jin Lifeng, Yu Dong, and Gildea1 Daniel. 2022. Hierarchical context tagging for utterance rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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 (EMNLP)*, pages 2846–2857, Online. Association for Computational Linguistics.

Scott Martin, Shivani Poddar, and Kartikeya Upasani. 2020. MuDoCo: Corpus for multidomain coreference resolution and referring expression generation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 104–111, Marseille, France. European Language Resources Association.

Zhu Feng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019a. 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, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1824–1833. Association for Computational Linguistics.

Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019b. 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 (EMNLP-IJCNLP)*, pages 1824–1833, Hong Kong, China. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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 (EMNLP-IJCNLP)*, pages 4547–4557, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Michael Regan, Pushpendre Rastogi, Arpit Gupta, and Lambert Mathias. 2019. A dataset for resolving referring expressions in spoken dialogue via contextual query rewrites (cqr). *arXiv preprint arXiv:1903.11783*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019a. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 22–31. Association for Computational Linguistics.

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

Bo-Hsiang Tseng, Shruti Bhargava, Jiarui Lu, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Lin Li, and Hong Yu. 2021. CREAD: Combined resolution of ellipses and anaphora in dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3390–3406, Online. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

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, 23-27 May 2022*, pages 8047–8051. IEEE.

A Examples of Datasets

Dataset	Description and Examples
MuDoCo	Daily conversation of six domains.
Train 2.39k Dev 0.29k Test 0.30k Ave Len 73.43 % RW 26.68	**Context** **A:** put me on active now . **B:** you are active now . **A:** did i miss any calls or messages here ? **B:** mila called yesterday at 1 am . **Current Utterance** **A:** is she on now ? **Ground Truth** **A:** is mila on now ?
CQR	Task-oriented dialogue between a user and an agent.
Train 0.52k Dev 0.06k Test 0.06k Ave Len 143.70 % RW 98.38	**Context** **A:** What gas stations are here ? **B:** There is a Chevron . **A:** That ' s good ! Please pick the quickest route to get there and avoid all heavy traffic ! **B:** Taking you to Chevron . **Current Utterance** **A:** What is the address ? **Ground Truth** **A:** What is the address of the gas station Chevron ?
REWRITE	Chinese dataset of 3-turn dialogues.
Train 16.00k Dev 2.00k Test 2.00k Ave Len 36.85 % RW 99.98	**Context** **A:** 能给我签名吗 (Could you give me signature?) **B:** 出专辑再议 (Wait until the album is released.) **Current Utterance** **A:** 我现在就要 (I want it now.) **Ground Truth** **A:** 我现在就要签名 (I want signature now.)
RES	Chinese dataset of 200K multi-turn conversations in open-domain.
Train 193.77k Dev 5.10k Test 5.10k Ave Len 68.38 % RW 60.00	**Context** **A:** 今天买了一堆桌游有爱玩的可以一起 (Today, I bought a lot of board games. Those who like to play can join me.) **B:** 我比较喜欢卡卡颂和现代艺术 (I prefer Kakason and modern art.) **A:** 听说过不过没买 (I heard about it, but I didn't buy it.) **B:** 我有 (I have it.) **Current Utterance** **A:** 一起啊 (Let's play together.) **Ground Truth** **A:** 一起玩桌游啊 (Let's play board games together.)
CANARD	Teacher and student talking about news or a person.
Train 16.88k Dev 1.79k Test 2.96k Ave Len 429.77 % RW 92.91	**Context** **A:** anna politkovskaya **B:** the murder remains unsolved , 2016 **Current Utterance** **A:** did they have any clues ? **Ground Truth** **A:** did investigators have any clues in the unresolved murder of anna politkovskaya ?

Table 10: Information and examples of 4 datasets.

The brief descriptions, statistics and samples of the datasets are shown in Table 10.

B Cases in CANARD

Table 11 shows some specific examples of rewriting using our model and other baselines. The examples of predicting results of our model, HCT, RUN and T5-small on CANARD dataset are shown from top to bottom. HCT tends to copy the predicted span directly from the context. From the first example, we can find that HCT predicts the correct position of coreference in the current sentence, but finds the wrong span. From the second example, we can see that RUN's edit based model duplicates the span from the context. Our model uses T5 to find the corresponding span from the context, which is significantly stronger than RUN and HCT.

The third example shows the shortcomings of our model. Compared with the end-to-end T5-small model, the first step of our framework failed to predict the need to insert words between "write" and "real", so the second step could not fill in the correct answer. This shows the inherent defect of the 2-step framework, that is, the result of the second step depends on that of the first step, and there is a certain gap.

Context	**A:** betsy devos **B:** school vouchers **A:** what are the school vouchers ? **B:** would allow students to attend private schools with public funding .
Current	**A:** how do people get them ?
Gold	**A:** how do people get the school vouchers ?
Ours-sup	**A:** how do people get school vouchers ?
HCT	**A:** how do people get private ?
Context	**A:** anna ella carroll **B:** 1850s political career **A:** what made anna get into politics ? **B:** carroll joined the american party (the know nothing party) following the demise of the whigs .
Current	**A:** where was she when she started the american party ?
Gold	**A:** where was anna ella carroll when she started the american party ?
Ours-sup	**A:** where was anna ella carroll when she started the american party ?
RUN	**A:** where was anna ella anna ella carroll when she started the american party ?
Context	**A:** real love (beatles song) **B:** early origins
Current	**A:** who originally wrote real love ?
Gold	**A:** who originally wrote beatles song real love ?
Ours-sup	**A:** who originally wrote real love ?
T5-small	**A:** who originally wrote the beatles song real love ?

Table 11: More typical examples extracted from the prediction results on CANARD.

C Cases in CQR and MuDoCo

As a supplement to case study, we provide more cases from CQR and MuDoCo here in Table 12.

Context (**CQR**)	**A:** What gas stations are here ? **B:** There is a Chevron .
Current	**A:** That ’ s good ! Please pick the quickest route to get there and avoid all heavy traffic !
Gold	**A:** That ’ s good ! Please pick the quickest route to get to the gas station Chevron and avoid all heavy traffic !
Ours-sup	**A:** That ’ s good ! Please pick the quickest route to get to the gas station Chevron and avoid all heavy traffic !
RUN	**A:** a chevron ’ s good ! please pick the quickest route to the there and avoid all heavy chevron traffic !
Context (**CQR**)	**A:** where is the closest grocery store **B:** We are 4 miles away from Whole Foods and from Safeway : which one do you prefer ? **A:** Safeway . **B:** Safeway is located at 452 Arcadia Pl .
Current	**A:** Pick the quickest route to go there and send the info on my screen please
Gold	**A:** Pick the quickest route to go to the grocery store Safeway 4 miles away at 452 Arcadia Pl and send the info on my screen please
Ours-sup	**A:** Pick the quickest route to go to the grocery store Safeway 4 miles away at 452 Arcadia Pl and send the info on my screen please
T5-small	**A:** Pick the quickest route to go to the grocery store Safeway 4 miles away at 452
Context (**MuDoCo**)	**A:** if ray or ron call do not answer . **B:** i will not answer calls from ray or ron . **A:** add rob to that list too . **B:** you do not wish to receive calls from ray , ron or rob today . correct ? **A:** yep , but if roy calls answer that ! **B:** i will only answer if roy calls not the others .
Current	**A:** yes he is the only one i want to talk to today .
Gold	**A:** yes roy is the only one i want to talk to today .
Ours-sup	**A:** yes roy is the only one i want to talk to today .
HCT	**A:** yes rob is the only one i want to talk to today .
Context (**MuDoCo**)	**A:** who is that calling me now ? **B:** the call is from eric , alex , and kyle .
Current	**A:** decline the call and tell them all that i will call them back .
Gold	**A:** decline the call and tell eric , alex , and kyle that i will call them back .
Ours-sup	**A:** decline the call and tell eric , alex , and kyle that i will call them back .
T5-small	**A:** decline the call and tell eric, alex, and kyle all

Table 12: Examples extracted from the prediction results of CQR and MuDoCo. Red words shows the different parts between our framework and baseline.