Incomplete Utterance Rewriting as Sequential Greedy Tagging

Yunshan Chen
SF Technology Co., Ltd.
chenyunshan.ai@gmail.com

Abstract

The task of incomplete utterance rewriting has recently gotten much attention. Previous models struggled to extract information from the dialogue context, as evidenced by the low restoration scores. To address this issue, we propose a novel sequence tagging-based model, which is more adept at extracting information from context. Meanwhile, we introduce speaker-aware embedding to model speaker variation. Experiments on multiple public datasets show that our model achieves optimal results on all nine restoration scores while having other metric scores comparable to previous state-of-the-art models. Furthermore, benefitting from the model's simplicity, our approach outperforms most previous models on inference speed.

1 Introduction

Recent years have witnessed increasing attention in dialogue systems mainly due to its promising potential for applications like virtual assistants or customer support systems (Hauswald et al., 2015; Debnath et al., 2018). However, studies (Carbonell, 1983) show that users of dialogue systems tend to use incomplete utterances which usually omit (a.k.a. ellipsis) or refer back (a.k.a. co-reference) to the concepts that appeared in previous dialogue contexts. (also known as non-sentential utterances, (Fernández et al., 2005). Thus, dialogue systems must understand these incomplete utterances to make appropriate responses.

To tackle the problem, the task of **I**ncomplete **U**tterance **R**ewriting(**IUR**, also known as context rewriting) (Su et al., 2019; Pan et al., 2019; Elgohary et al., 2019), which aims to rewrite an incomplete utterance into an utterance that is semantically equivalent but self-contained to be understood without context, has recently become an increasing focus of NLP research. As depicted in Table 1, the incomplete utterance u3 not only omits the subject "深圳"(Shenzhen), but also refers to the semantics of "经常阴天下雨"(always raining) via the pronoun "这样"(this). The downstream dialogue model only needs to take the last utterance by explicitly recovering the dropped information into the latest utterance. Thus, the burden of long-range reasoning can be primarily relieved, making the downstream dialogue modeling more accurate.

Turn	Utterance with Translation
u_1 (A)	深圳最近天气怎么样? (How is the recent weather in Shenzhen?)
u_2 (B)	最近经常阴天下雨。 (It is always raining recently.)
u_3 (A)	冬天就是这样的。 (Winter is like this.)
u_3'	深圳冬天就是经常阴天下雨。 (It is always raining in winter Shenzhen.)

Table 1: An example dialogue between speaker A and B, including the context utterances (u_1, u_2), the incomplete utterance (u_3) and the rewritten utterance (u_3').

The previous top work on building IUR model mainly includes generation-based methods and tagging-based methods. Generation-based solution (Su et al., 2019; Pan et al., 2019; Elgohary et al., 2019) consider this task as a standard text-generation problem, adopting a sequence-to-sequence model with a copy mechanism (Gulcehre et al., 2016; Gu et al., 2016; See et al., 2017). However, those methods generate the rewritten utterance from scratch, which introduces an over-large search space and neglects the critical trait that the main structure of a rewritten utterance is always the same as the incomplete utterance.

In order to break through those limitations, tagging-based approach (Liu et al., 2020; Hao et al., 2021; Jin et al., 2022; Zhang et al., 2022; Wang et al., 2022) was proposed. For specifically, here we consider models like RUN (Liu et al., 2020) as a tagging-based method. Its semantic segmentation

Findings of the Association for Computational Linguistics: ACL 2023, pages 7265–7276
July 9-14, 2023

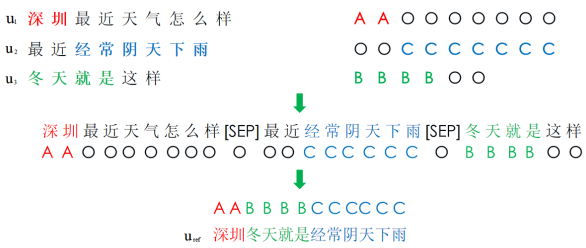

Figure 1: The illustration of the main learning task of the sequence tagging model **SGT**(**S**equential **G**reedy **T**agging). We adopt the same example dialogue from Table 1. Considering SGT is position-dependent, and word order between Chinese and English is different, the corresponding English utterance is not provided, which is the same for Figure 3.

task can be analogous to the sequence tagging task. The main difference is that the semantic segmentation task is tagging in two-dimensional coordinates, while the sequence annotation task is tagging in one-dimensional coordinates.

The previous top tagging-based approach generally formalizes IUR as learning the edit operation and corresponding location. The tagging-based approach enjoys a smaller search space than the generation-based approach and can better utilize the information that the main structure of a rewritten utterance is always the same as the incomplete utterance.

Despite their success, existing approach that learning edit operation and the corresponding location has difficulty handling situations where multiple inserts correspond to one position. Moreover, models like RUN adopt a heavy model that takes ten convolution layers in addition to the BERT encoder, which will increase its training time and slows down its infer speed. More critically, although BERT (Devlin et al., 2019) has shown to be powerful in extracting information, the generally low restoration scores prove that previous BERT-based models are ineffective in extracting the information needed for IUR from historical utterances. Finally, the experimental results of SA-BERT (Gu et al., 2020) demonstrate that explicitly modeling speaker changes has a specific enhancement effect on modeling multi-turn dialogue tasks. The previous approach did not model this critical information.

To address these issues, we propose a novel sequence tagging model named **SGT**(**S**equential **G**reedy **T**agging), which is not based on learning editing operations and can significantly improve the restoration score and inference speed. Our solution was derived from the following thinking: First, we consider that in the dialogue process, any complete utterance is composed by only of a few fragments. For example, "I love you" includes three components: subject, verb, and object. Even if it is expanded with modifications and qualifications, its composition is still minimal. Based on this insight, we thought it would be possible to build a model to identify the fragments and their order from dialogue history to form a target completed utterance. And then, splice those fragments together in sequence and get the complete utterance. Meanwhile, in order to keep the number of fragments constituting the target rewritten utterance relatively small, we adopt the greedy tagging strategy. Our model will identify all the fragments and their order required to form a completed utterance; each fragment is the longest fragment found in the given order. We might as well call this fragment **GLCS** (**G**reedy **L**ongest **C**ommon **S**ubsequence). Specifically, we use the tag type to represent the order of GLCS for composing the target rewrite utterance, For example, the first GLCS that constitutes a rewritten utterance would be tagged as A, and the second is B, the third is C, and so on. In the above manner, we converted IUR into a simple sequence tagging task, as illustrated in Figure 1. After the model has identified all GLCSs from the dialogue history through this strategy, the target rewritten utterance can be obtained by splicing each GLCS in alphabetical order according to its tag.

Furthermore, we introduce speaker-aware embedding to model the speaker changes in different rounds. Finally, to better perceive the boundaries of each tagging mention, we add two simple losses in addition to the sequence labeling loss.

In summary, our contributions are as follows:

1. We proposed SGT, a novel paradigm to model IUR. Due to the simplicity and effectiveness of modeling, our approach can fully utilize the sequence labeling capabilities of BERT to extract information from historical utterances and thus restore incomplete utterances with more accuracy. Experiments on several datasets show that our method significantly improved the ability to extract mentions from context, which are argued to be harder to copy by (Pan et al., 2019).

2. To the best of our knowledge, we are the

first to introduce speaker-aware embedding to model IUR.

3. Finally, benefit from the model's simplicity. Our inference speed is faster than most previous models.

2 Related Work

Earlier efforts (Su et al., 2019; Elgohary et al., 2019) treated dialogue utterance rewriting as a common text generation problem and integrated seq-to-seq models with copy mechanisms to model this task. Later work (Pan et al., 2019; Zhou et al., 2019; Huang et al., 2021) explore task-specific features for additional gains in performance. For example, (Pan et al., 2019) adopt a pipeline-based method. The idea is to detect keywords first and then append those words to the context and adopt a pointer generator that takes the output of the first step to produce the output. However, this two-step method inevitably accumulates errors.

SRL (Xu et al., 2020) trains a semantic labeling model to highlight the central meaning of keywords in the dialogue as a sort of prior knowledge for the model. To obtain an accurate SRL model for dialogues, they manually annotate SRL information for more than 27,000 dialogue turns, which is costly and time-consuming.

RUN (Liu et al., 2020) convert this task into a semantic segmentation problem, a significant task in computer vision. In particular, their model generates a word-level edit matrix, which contains the operations of insertion and substitution for each original utterance. Rather than word embeddings, RAU (Zhang et al., 2022) directly extracts ellipsis and co-reference relationships from Transformer's self-attention weighting matrix and edits the original text accordingly to generate complete utterances. RUN++ (Wang et al., 2022) Introduce contrastive learning and keyword detection tasks to model the problem jointly. Both RAU and RUN++ make significant improvements in most metrics on several datasets. Although some additional effective strategies exist. It is still in the same paradigm as RUN, learning edit matrix by cast IUR as a semantic segmentation task.

RAST (Hao et al., 2021) is the first work to convert dialogue utterance rewriting into a sequence tagging task. It takes experimentation to prove that most models for this task suffer from the robustness issue, i.e., performance drops when testing on a different dataset. By contrast, RAST is more robust than the previous works on cross-domain situations. Moreover, this work design additional reinforcement learning task to improve fluency. Despite all these efforts, its overall in-domain performance still lags behind methods that learn edit operation matric (Liu et al., 2020).

To better enhance pre-trained language models for multi-turn response selection in retrieval-based chatbots. A model named **S**peaker-**A**ware BERT (SA-BERT) (Gu et al., 2020) proposed to make the model aware of the speaker's changed information, which is an essential and intrinsic property of multi-turn dialogues.

Although RAST has a different learning paradigm from works that learn edit matrix, it still tries to learn the edit operation and corresponding location by sequence tagging. As mentioned before, our method is sequence tagging-based but takes an entirely new paradigm that would not learn edit operations. Besides, inspired by SA-BERT (Gu et al., 2020), we introduce speaker embedding to this task. Finally, we introduce two simple sequence labeling tasks to model this problem jointly.

3 Methodology

3.1 Task Definition

Here we give the formal definition of how we model the IUR problem with the SGT approach. Taking all history utterances $H = (U_1, U_2, ..., U_n)$ as input, SGT aims to learn a function to rewrite U_n to R: $f(H) \rightarrow R$. R is the target rewritten utterance in the infer stage. In particular, U_n is the last utterance of all history utterances and the utterance that needs to be rewritten in the IUR task. R is the reference rewritten utterance U_{ref} in the training phase and the target rewritten utterance in the inference phase.

3.2 Model Architecture

Figure 2 shows the overall architecture of our model.

Contextual Encoder Since pre-trained language models have been proven to be effective in many NLP tasks, our experiment employs BERT (Devlin et al., 2019) to be encoder. For a fair comparison, we take the same BERT-base encoder as the previous sota work (e.g., RUN, RAU, RUN++) to represent each input. Concretely, given input token list $H = (x_1, x_2, \cdots, x_M)$ which concatenated by all utterances of dialogue history and inserted a special

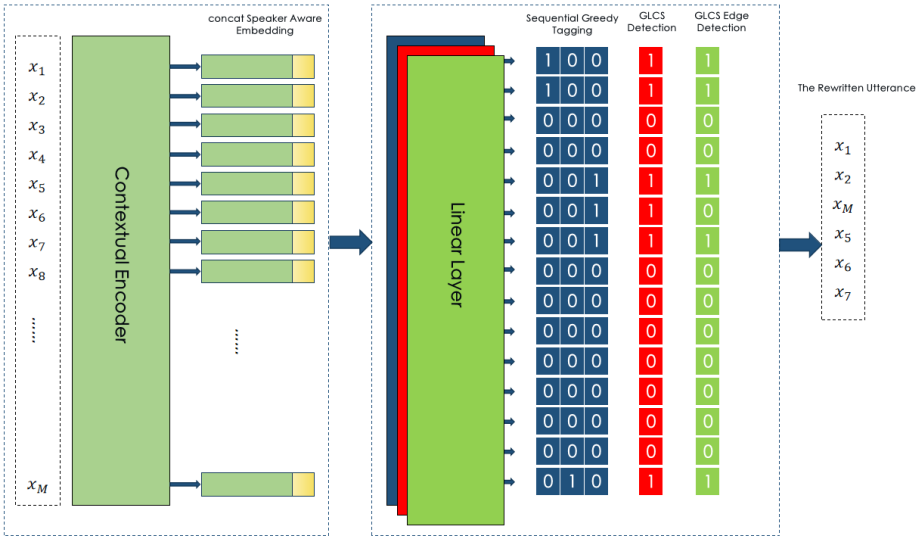

Figure 2: This figure depicts the SGT model's overall structure, mainly contextual embedding, speaker-aware embedding, and three linear layers for the SGT main task and two additional tasks, respectively. Among the three learning tasks, the dark blue part represents the SGT task, the red part represents the **G**LCS **D**etection task(GD), and the green part represents the **G**LCS **E**dge **D**etection task (GED).

token [SEP] between each utterance for separate utterances in different turns. The BERT encoder is firstly adopted to represent the input with contextualized embeddings and the calculation of this part is defined as:

$$E = (\mathbf{e}_1, \cdots, \mathbf{e}_M) = BERT\,(H) \qquad (1)$$

Speaker Aware Embedding To distinguish utterances between different speakers, our approach stitches a one-dimensional one-hot vector at the hidden dimension with the output representation of the BERT encoder. This design is based on two considerations. On the one hand, most of the dialogue in the dataset is back-and-forth conversations between only two people. On the other hand, adding speaker embedding at the input layer and performing domain adaptation like SA-BERT will make the encoder different from the BERT-based model, which would contradict the fair comparison conditions we assumed earlier in paragraph 3.2. The calculation of this part is defined as follows::

$$EA = Concat\,(Dropout\,(E)\,, SA) \qquad (2)$$

In the above equation, $E \in R^{M\times 768}$ is the output representation from the contextual encoder. $SA \in R^{M\times S}$ denotes the speaker-aware embedding. We concatenate E and SA alongside its hidden dimension to get $EA \in R^{M\times(768+S)}$.

Sequential Greedy Tagging Our main task is sequential greedy tagging, this can be generally defined as:

$$P_{sgt} = f(H) \qquad (3)$$

Specifically, $H = (x_1, x_2, \cdots, x_M)$ is the input token list that concatenated by the dialogue's history utterances. The model learns a mapping function f to predict from H to the token-level sequence labeling matrix $P_{sgt} \in R_{MXN}$, where M is the token number of sequence H, and N is the number of tag types. The objective function is defined as:

$$L_{sgt} = \frac{1}{M \times N} \sum_{i=0}^{M\times N} CE\left(P^i_{sgt}, Y^i_{sgt}\right) \qquad (4)$$

Where Y^i_{sgt} is the target type of the i-th sample at the token level. CE is the notation of cross-entropy loss which is the same for both equations 5 and 6.

GLCS Detection and GLCS Edge Detection To better lock in the span of target GLCS needed to make up the rewritten utterance, we introduced multi-task learning.

Firstly, as depicted by the red components on the right side of Figure 2, the **G**LCS **D**etection module (GD) is a binary classification task to distinguish whether a token should belong to a target GLCS. The module GD outputs $P_{gd} \in R_{MX1}$. LD is essentially a sequence tagging problem, and the loss function of the GLCS detection is as follows:

$$L_{gd} = \frac{1}{M} \sum_{i=0}^{M} CE\left(P^i_{gd}, Y^i_{gd}\right) \qquad (5)$$

Y^i_{gd} is the golden mentions label of the i-th sample. P^i_{gd} is the predicted mentions label of the i-th sample.

Secondly, as depicted by the green components on the right side of Figure 2, the **G**LCS **E**dge **D**etection module (GED) is a binary classification task with a structure similar to GD. Specifically, a target that consists of a single token or only two tokens will be marked throughout as 1; only tokens at its start position and end position will be marked as 1 when more than three tokens, left with the others as 0. The loss function of the GED is as follows:

$$L_{ged} = \frac{1}{M}\sum_{i=0}^{M} CE\left(P^i_{ged}, Y^i_{ged}\right) \tag{6}$$

Y^i_{ged} is the golden mentions label of the i-th sample. P^i_{ged} is the predict mentions label of the i-th sample.

Final Learning objectives Finally, we combine all tasks and train them simultaneously by taking the summation of all loss functions, and the final loss function is shown below:

$$L_{final} = L_{gd} + L_{ged} + L_{sgt} \tag{7}$$

3.3 Data Construction

The construction of the training data for the SGT task is shown in Figure 3. First, in step S_1, we make $U^{(1)}_{ref} = U_{ref}$, then find the LCS between each history utterance and U^1_{ref} separately. Also, this LCS needs to satisfy being a prefix of $U^{(1)}_{ref}$. After step S_1, we can get the first GLCS "深圳"(Shenzhen), and we set the label of its corresponding position to "AA." Then, in step S_2, we make $U^{(2)}_{ref}$ = ($U^{(2)}_{ref}$ *remove the prefix* "深圳"(Shenzhen)). Performing the same GLCS search process, we can obtain the second GLCS "冬天就是"(winter is) and set its label as "BBBB." Analogously, we can get the third GLCS "经常阴天下雨"(always cloudy and raining) and set its label as "CCCCCC" at Step S_3. Finally, the historical utterances are stitched together as the input of the SGT task. The corresponding labels obtained from steps S_1, S_2, and S_3 are used as the labels of the sequence labeling task.

Points need to be clarified: (i) **Granularity** The token sequence is char level for Chinese and word level for English and numbers, both in the GLCS

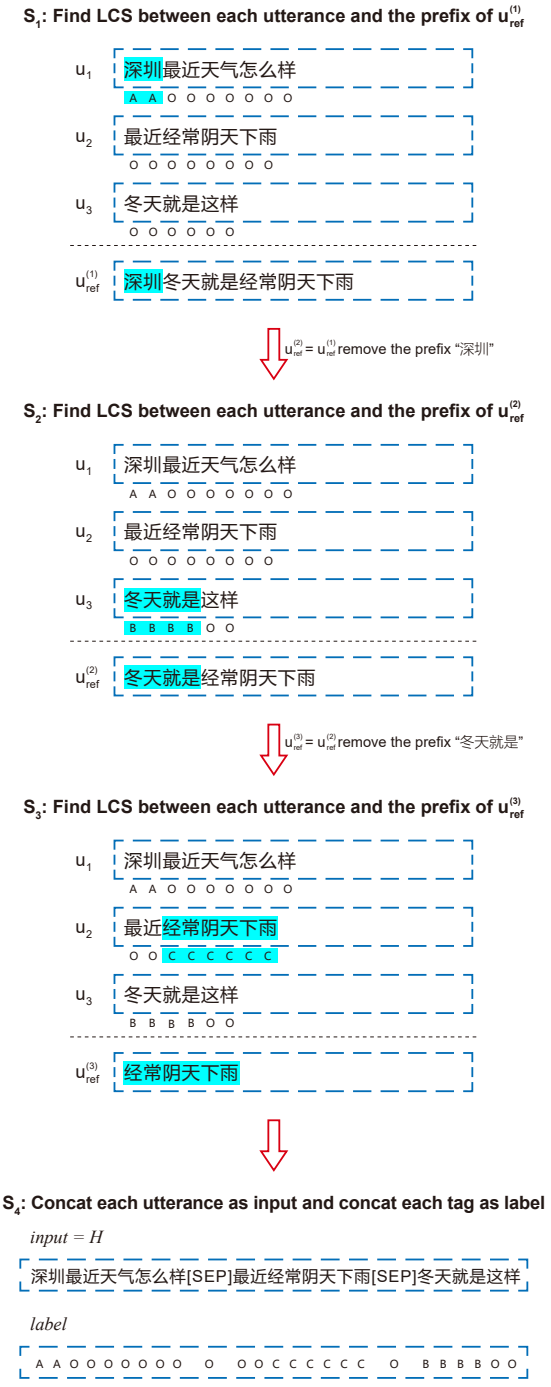

Figure 3: This figure depicts how the training data for the sequence tagging task required by SGT can be generated from the original dataset's history utterances and the reference utterance.

matching phase of S_1, S_2, and S_3 and in the training phase of data obtained from S_4, which is the same as RUN; (ii) **Duplicate Matching** In case of duplicate matches, e.g., if U_1 and U_2 have the same desired GLCS, the principal is the latter, the better.

4 Experiments

In this section, we conduct through experiments to demonstrate the superiority of our approach.

Datasets We conduct experiments on three public datasets across different domains: Chinese datasets in open-domain dialogues: MULTI (Pan et al., 2019) and REWRITE (Su et al., 2019) , English Task-Oriented Dialogue TASK (Quan et al., 2019). For a fair comparison, We adopt the same data split for these datasets as our baselines. The statistics of these datasets are displayed in Table 2.

	MULTI	REWRITE	TASK
Language	Chinese	Chinese	English
Train	194K	18K	2.2K
Dev	5K	2K	0.5K
Test	5K	NA	NA
Avg. C len	25.5	17.7	52.6
Avg. Q len	8.6	6.5	9.4
Avg. R len	12.4	10.5	11.3

Table 2: Statistics of different datasets. NA means the development set is also the test set. "Ques" is short for questions, "Avg" for short for average, "len" for length, "C" for context utterance, "Q" for current utterance, and "R" for rewritten utterance.

Baselines To prove the effectiveness of our approach, we take the State-of-the-art models as strong baselines including SRL (Xu et al., 2020), SARG (Huang et al., 2021), PAC (Pan et al., 2019), RAST (Hao et al., 2021), T-Ptr-λ (Su et al., 2019), RUN (Liu et al., 2020) and RUN++ (Wang et al., 2022) .

Evaluation We employ credible automatic metrics to evaluate our approach. As in literature (Pan et al., 2019), we examine SGT using the widely used automatic metrics BLEU, ROUGE, EM and Restoration F-score. (i) **BLEU**$_n$ (**B**$_n$) evaluates how similar the rewritten utterances are to the golden ones via the cumulative n-gram BLEU score (Papineni et al., 2002). (ii) **ROUGE**$_n$ (**R**$_n$) measures the n-gram overlapping between the rewritten utterances and the golden ones, while **ROUGE**$_L$ (**R**$_L$) measures the longest matching sequence between them (Lin, 2004). (iii) **EM** stands for the exact match accuracy, which is the strictest evaluation metric. (iv) **Restoration Precision**$_n$, **Restoration Recall**$_n$ and **Restoration F-score**$_n$ ($\mathcal{P}_n$, $\mathcal{R}_n$, $\mathcal{F}_n$) emphasize more on words from dialogue context which are argued to be harder to copy (Pan et al., 2019). Therefore, they are calculated on the collection of n-grams that contain at least one word from context utterance. As validated by Pan et al. (2019), above automatic metrics are credible indicators to reflect the rewrite quality.

Implementation Our implementation was based on PyTorch (Paszke et al., 2019) and fastNLP (Xipeng Qiu, 2018). In practice, we adopt the exact connection words setting with RUN and append the list of connection words to the head of H, as part of it. Considering that only two speakers are in the datasets related to our experiments, we set the hidden_size of SA to 1. For encoding different tagging types, We choose **IO** encoding, the simplest tag encoding schema, which tags each token as either being in (I-X) a particular type of named entity type X or in no entity (O). Since the distribution of tag types is severely unbalanced (e.g. (O) accounts for more than 81% on MULTI), we employed weighted cross-entropy loss and tuned the weight on development sets. We used Adam (Kingma and Ba, 2014) to optimize our model and set the learning rate as 2e-5. We set the dropout rate as 0.3 for the dropout operation on the equation 2. For a fair comparison, the BERT used in our model is BERT-base which is the same as our baselines.

4.1 Model Comparison

Table 3, Table 4, and Table 5 show the experimental results of our approach and baselines on MULTI and REWRITE. As shown, our approach greatly surpasses all baselines on practically all restoration scores significantly. Taking MULTI as an example, our approach exceeds the best baseline RUN++(PCL) on restoration score by a significant margin, reaching a new state-of-the-art performance on almost all restoration metrics. Our approach improves the previous best model by 9.79 points and 9.89 points on restoration $\mathcal{F}_3$ and $\mathcal{F}_2$, respectively. Furthermore, our approach reaches comparable performance on other auto metrics. As demonstrated by the result of REWRITE, our approach achieves comparable performance

Model	$\mathcal{P}_1$	$\mathcal{R}_1$	$\mathcal{F}_1$	$\mathcal{P}_2$	$\mathcal{R}_2$	$\mathcal{F}_2$	$\mathcal{P}_3$	$\mathcal{R}_3$	$\mathcal{F}_3$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{R}_1$	$\mathbf{R}_2$
SRL	NA	NA	NA	NA	NA	NA	NA	NA	NA	85.8	82.9	89.6	83.1
T-Ptr-λ (n_beam=5)	NA	NA	51.0	NA	NA	40.4	NA	NA	33.3	90.3	87.7	90.1	83.0
PAC (n_beam=5)	70.5	58.1	63.7	55.4	45.1	49.7	45.2	36.6	40.4	89.9	86.3	91.6	82.8
SARG (n_beam=5)	NA	NA	62.3	NA	NA	52.5	NA	NA	46.4	91.4	88.9	91.9	85.7
RAST	NA	NA	NA	NA	NA	NA	NA	NA	NA	89.7	88.9	90.9	84.0
RUN	73.2	64.6	68.8	59.5	53.0	56.0	50.7	45.1	47.7	**92.3**	**89.6**	92.4	85.1
RUN++(PCL)	NA	NA	71.1	NA	NA	59.1	NA	NA	51.1	92.1	89.4	92.6	**86.2**
SGT(Ours)	**75.0**	**67.5**	**71.1**	**73.1**	**65.3**	**69.0**	**64.7**	**57.5**	**60.9**	92.1	89.0	**92.7**	85.3

Table 3: Reuslts on MULTI. All models except T-Ptr-λ are initalized from pretrained Bert-base-Chinese model. All results are extracted from the original papers. The final line is the result of our complete model. A bolded **number** in a column indicates a sota result against all the other approach, whereas underline numbers show comparable performances. Both are same for Table 4&5.

.

Model	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	**EM**	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_4$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$
SRL	NA	NA	NA	60.5	89.7	86.8	77.8	91.8	85.9	90.5
RAST	NA	NA	NA	63.0	89.2	88.8	86.9	93.5	88.2	90.7
RUN	89.3	81.9	76.5	67.7	93.5	91.1	86.1	95.3	90.4	94.3
RUN++(PCL)	89.8	83.2	78.2	**69.0**	93.7	91.5	**87.0**	95.6	**91.0**	**94.6**
SGT(Ours)	**91.0**	**89.8**	**85.1**	67.4	**94.9**	**92.2**	86.8	**96.4**	90.8	93.8

Table 4: Results on REWRITE. All models are initialized from pretrained Bert-base-Chinese model. All baseline results are extracted from the RUN++ (Wang et al., 2022). The final line is the result of our complete model.

Model	**EM**	$\mathbf{B}_4$	$\mathcal{F}_1$
Ellipsis Recovery †	50.4	74.1	44.1
GECOR 1 †	68.5	83.9	66.1
GECOR 2 †	66.2	83.0	66.2
RUN	70.6	86.1	68.3
SGT(Ours)	**71.1**	**86.7**	**85.0**

Table 5: The experimental results on TASK. † Results from Quan et al. (2019). RUN and SGT are initialized from pretrained Bert-base model, which are same for Table 6.

on the $\mathbf{B}_4$, $\mathbf{R}_2$, and $\mathbf{R}_L$ scores and a new state-of-the-art performance on $\mathbf{B}_1$ and $\mathbf{R}_1$ scores. Even for the most strict metric EM on REWRITE, our approach reached comparable performance with RUN, demonstrating the comprehensive ability of our model. Besides, our approach achieves better results against all baselines on TASK, as depicted in Table 5. Specifically, we achieve state-of-the-art performance on the EM score and exceed the previous best model by 16.7 points on the restoration $\mathcal{F}_1$ score. Finally, the combined performance of our model on the three datasets above demonstrates that our model can perform well on datasets with varied languages and tasks.

4.2 Closer Analysis

We conduct a series of experiments to analyze our model thoroughly. First, we conduct a detailed ablation study to validate the efficacy of the components in our model. Then, in the same run-time setting, we compare the inference speed of our model to that of representative baselines.

Ablation Study By analyzing table 6, we can find that "w/o sa","w/o gd" or "w/o ged" basically hurts the effect of the model, and these can initially corroborate that each of these modules is beneficial to our model.

Meanwhile, we can find that "w/o gd+ged" significantly reduces $\mathcal{R}_3$, indicating that these two subtasks are very helpful for discovering the potential target GLCS. Further, we find that although removing "sa" alone has little effect on the restoration score, comparing the results of removing "gd+ged" and removing "sa+gd+ged" reveals that the fit with the missing speaker-aware information significantly reduces the restoration score. The $\mathcal{F}_3$ decreases from 74.7 to 71.8, which indicates that the information of different speakers or rounds is

Variant	$\mathcal{P}_3$	$\mathcal{R}_3$	$\mathcal{F}_3$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{R}_L$
SGT	81.9	71.7	76.5	94.5	91.4	94.5
SGT w/o (sa)	80.8	71.3	75.8	94.1	91.2	94.5
SGT w/o (gd)	80.9	71.6	76.0	94.1	90.9	94.4
SGT w/o (ged)	81.5	70.7	75.8	94.0	91.2	94.4
SGT w/o (gd+ged)	82.0	68.5	74.7	93.6	90.6	94.1
SGT w/o (sa+gd+ged)	79.4	65.5	71.8	93.6	90.3	93.8
RUN	70.7	45.7	55.5	91.5	89.4	93.7

Table 6: The ablation results on the development set of TASK. "SGT" denotes our complete model. "w/o sa" indicates without the speaker-aware embedding. "w/o gd" means that remove GLCS detection task from our multi-task learning. "w/o ged" means that remove GLCS Edge detection task from our multi-task learning. Other remaining variants can be deduced in the same manner.

crucial to extract the target GLCS correctly, and combining "sa" embedding and "gd+ged" subtasks can significantly improve the model's ability to obtain the target GLCS fragments from the context.

Finally, we find that even though the absence of the three critical components "sa+gd+gd" leads to an overall decrease in model performance, our model still achieves a better restoration score than the RUN model, which further validates the effectiveness of our sequential greedy tagging learning strategy for modeling and solving UIR problems.

Model	$\mathbf{B}_4$	$\Delta\mathbf{B}_4$	**Latency**	**Speedup**
L-Gen	73.6	0.0	82 ms	1.00 ×
L-Ptr-Gen	75.4	+1.8	110 ms	0.75 ×
T-Gen	62.5	-11.1	322 ms	0.25 ×
T-Ptr-Gen	77.6	+4.0	415 ms	0.20 ×
RUN	86.2	+12.6	71 ms	1.15 ×
SGT	86.8	+13.2	51 ms	1.60 ×

Table 7: The comparison of inference speeds between SGT and baselines. We set the beam size parameter to 4 for approaches that need the beam search method, which is not relevant to RUN and SGT. Meanwhile, we did not do inference performance measurements on RUN++ and other RUN-based models with comparable inference structures, considering they are theoretically nearly identical to RUN. Latency is calculated as the time it takes to produce a single phrase without data batching, averaged over the REWRITE development set. All models are built in PyTorch and run on a single NVIDIA V100.

Inference Speed As shown in Table 7, both SGT and RUN significantly outperform traditional generation algorithms regarding inference speed and $\mathbf{B}_4$ score. At the same time, the most time-consuming computation of SGT in the inference phase, except for the BERT encoder, is only one layer of a linear transformation, which dramatically saves the inference time compared with RUN, which has U-net (Ronneberger et al., 2015) structures after the context encoder. Therefore, we can see that the inference time of SGT is significantly less than that of RUN. The latency of a single rewriting task is reduced by 20ms, while the $\mathbf{B}_4$ score slightly better.

5 Conclusion

In this paper, we convert the IUR problem into a simple sequence tagging task, SGT. The simplicity and effectiveness of the modeling paradigm not only improve the inference speed and allow the pre-trained BERT encoder to fully exploit its widely validated information extraction ability which can significantly improve the restoration score and ensure that other metrics are competitive. We also introduced speaker-aware embedding to explicitly model speaker changes and verified that it has some improvement effect on the IUR task.

In the future, we will explore the following directions:

1. Adopt the GD task in this paper to extract essential fragments and then pick the best permutation of fragments with a language model or using a PAC-like pointer network for fragment integration to get rid of the problem of category imbalance is caused by representing the order with tag lists.

2. Combining SGT's efficient fragment extraction paradigm with generation.

Limitations

Although our model has made some progress, it still has some limitations. First of all, SGT uses the tag type to represent the connection order of GLCS fragments when forming a complete utterance, and the average statistics on the three datasets we used show that more than 99% of the complete utterance can be composed with less than three GLCS fragments. That will lead to situations that need to combine multiple GLCSs (e.g., more than 3) to form a complete utterance, which cannot be fully trained or fall into unbalanced tag categories. Second, like other tagging-based models, the fragments that make up the complete utterance must exist in history utterances or connection words, which does not work well for situations where it is necessary to combine context information and introduce new words to express their complete utterance.

Ethics Statement

We guarantee that the approach in our research is original and that the experimental results of the SGT model are reproducible. All the experimental data of the baseline model can be reproduced from the relevant open-source code or found in the cited paper. Finally, The list of authors of the submissions does not include any individuals who did not contribute substantially to work submitted.

Acknowledgements

First, we thank all the anonymous reviewers for their valuable comments. Moreover, we are grateful for all the previous work related to exploring the IUR task, which has inspired us a lot.

References

Jaime G Carbonell. 1983. Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 164–168.

Poulami Debnath, Shubhashis Sengupta, and Harshawardhan M Wabgaonkar. 2018. Identifying, classifying and resolving non-sentential utterances in customer support systems.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. *Can You Unpack That? Learning to Rewrite Questions-in-Context*.

Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2005. Using machine learning for non-sentential utterance classification. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 77–86.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

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.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Johann Hauswald, Michael A Laurenzano, Yunqi Zhang, Cheng Li, Austin Rovinski, Arjun Khurana, Ronald G Dreslinski, Trevor Mudge, Vinicius Petrucci, Lingjia Tang, et al. 2015. Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 223–238.

Mengzuo Huang, Feng Li, Wuhe Zou, Hongbo Zhang, and Weidong Zhang. 2021. Sarg: A novel semi autoregressive generator for multi-turn incomplete utterance restoration. In *AAAI*.

Lisa Jin, Linfeng Song, Lifeng Jin, Dong Yu, and Daniel Gildea. 2022. Hierarchical context tagging for utterance rewriting.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

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.

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.

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 (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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

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.

O. Ronneberger, P.Fischer, and T. Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer. (available on arXiv:1505.04597 [cs.CV]).

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.

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, Florence, Italy. Association for Computational Linguistics.

Zhihao Wang, Tangjian Duan, Zihao Wang, Minghui Yang, Zujie Wen, and Yongliang Wang. 2022. Utterance rewriting with contrastive learning in multi-turn dialogue. *arXiv preprint arXiv:2203.11587*.

Fudan NLP Xipeng Qiu. 2018. fastnlp, a lightweight framework for natural language processing (nlp). `https://github.com/fastnlp/fastNLP`.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic Role Labeling Guided Multi-turn Dialogue ReWriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639, Online. Association for Computational Linguistics.

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

Kun Zhou, Kai Zhang, Yu Wu, Shujie Liu, and Jingsong Yu. 2019. Unsupervised context rewriting for open domain conversation. 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 1834–1844, Hong Kong, China. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

8

☑ B1. Did you cite the creators of artifacts you used?
8

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.