Utterance Rewriting with Contrastive Learning in Multi-turn Dialogue

Anonymous ACL submission

Abstract

Context modeling plays a significant role in building multi-turn dialogue systems. In order to make full use of context information, systems can use Incomplete Utterance Rewriting(IUR) methods to simplify the multi-turn dialogue into single-turn by merging current utterance and context information into a self-contained utterance. However, previous approaches ignore the intent consistency between the original query and rewritten query. The detection of omitted or coreferred locations in the original query can be further improved. In this paper, we introduce contrastive learning and multi-task learning to jointly model the problem. Our method benefits from carefully designed self-supervised objectives, which act as auxiliary tasks to capture semantics at both sentence-level and token-level. The experiments show that our proposed model achieves state-of-the-art performance on several public datasets.

1 Introduction

With the development of single-turn dialogue modeling, remarkable progress has been achieved (Zhang and Zhao, 2021) in both question answering and open-domain response generation. However, in daily dialogues, users tend to omit or refer back to avoid repetitions. Research (Su et al., 2019) has shown that coreference and omission phenomenon exist in 33.5% and 52.7% of utterances, respectively, in pro-drop languages such as Chinese. Because users are capable of completing the simplified utterance by remembering the conversational history. Similarly, to equip the dialogue system with conversational memory, we would track the important history information through Dialogue States Tracking (DST). However, DST has three main problems: a) The volume of stored information is limited in long conversations. b) The stored information is pruned to avoid the redundancy of information. c) Other modules can rarely utilize the stored information in dialogue systems. Thus, To avoid problems of DST and boost the performance of dialogue understanding, recent researches propose simplifying multi-turn dialogue modeling into a single-turn problem by **I**ncomplete **U**tterance **R**erewring (**IUR**) (Kumar and Joshi, 2016; Pan et al., 2019; Su et al., 2019; Elgohary et al., 2019; Zhou et al., 2019; Liu et al., 2020; Zhou et al., 2019; Liu et al., 2020). Specifically, IUR is expected to recovering coreferred and omitted mentions in an incomplete utterance.

The examples from Table 1 correspond to the phenomena of coreference and omission respectively. "他"(he) from User_query is a coreference to "周杰伦"(Jay-Chou) in the first example and User_query from the second example omits the subject "上海"(Shanghai). In the two examples from Table 1, each User_query will be rewritten to User_query* by IUR. The dialogue system could modeling rewritten utterances more precisely without considering previous utterances. Besides, IUR is an extensible module that could be effortlessly integrated into different stages of a dialogue system, such as intent recognition or question answering tasks.

Previous work normally designs models in a two-stage way(Yin et al., 2018) including detecting the omitted or coreferred words and conducting the resolution task. However, it will introduce an accumulated error, i.e., a false detection leads to a false resolution. Recently, more and more researchers are focusing on designing end-to-end model architectures to solve IUR (Pan et al., 2019), (Su et al., 2019), as end-to-end models could avoid the problems of error accumulation and achieve better performance and speed. However, previous end-to-end models fail to explore the key traits of IUR task fully. For example, the natural assumption of IUR that the completed utterance should be semantically equivalent to the original dialogue is often neglected, while this trait could naturally be mod-

Turn	**Utterance***(Translation)*
User_context:	你喜欢周杰伦吗 *Do you like Jay-Chou*
System_context:	我喜欢周杰伦 *I like Jay-Chou*
User_query:	你喜欢他哪首歌 *Which song of him do you like*
User_query*:	你喜欢周杰伦哪首歌 *Which song of Jay Chou do you like*
User_context:	上海今天下雨吗 *Does it rain in Shanghai today*
System_context:	上海今天下雨 *It rains today in Shanghai*
User_query:	为什么最近总是下雨 *Why does it always rain recently*
User_query*:	为什么上海最近总是下雨 *Why does it always rain recently in Shanghai*

Table 1: Some examples of coreference and omission in daily life.

eled via a contrastive learning paradigm to boost the performance of IUR. Furthermore, we can still take advantage of two-stage methods while not being harmed by error accumulation with contrastive learning. In this work, we propose three simple yet effective designs to improve the performance of current IUR models. Our contributions are as follows:

1. As far as we know, we are the first to introduce contrastive learning into IUR.

2. We explore the key traits of IUR and modeling them in a multi-task learning paradigm.

3. Maintaining a fast inference speed, our approach achieves state-of-the-art performance on several Chinese datasets across different domains.

2 Related Work

Previous researches treat IUR as coreference resolution problem, adopting two-stage models to construct a detection-resolution pipeline (Yin et al., 2017, 2018). However, these methods often assume available golden syntactic parse trees, which are rare in real datasets. Recently, people have paid more attention to end-to-end models. The main-stream architectures could be classified into three categories: autoregressive(generation), semi-autoregressive and and non-autoregressive(sequence tagging). Most previous work model the problem as a standard autoregressive text generation task (Su et al., 2019; Quan et al., 2019; Elgohary et al., 2019). They adopt sequence-to-sequence models with copy mechanism to tackle the problem.

Besides, there are also work tackling IUR in a sequence-tagging paradigm RUN (Liu et al., 2020), RAST (Hao et al., 2020). RUN designs the model similarly to semantic segmentation in the computer vision domain, while RAST predicts target rewritten span for each token in the original query. These works are achieving the best performance on public datasets, as they further reduce the search space. In the meantime, avoiding the generation process means that these models have fast inference speed as they do not need beam search. Naturally, semi-auto-regressive model architectures are also explored by SARG(Huang et al., 2020).

Previous works also try to improve performance by designing specific tasks to utilize the traits of IUR. Through the review of previous work, we conclude the specific tasks into five perspectives: Pretraining, Keywords Detection, Search Space Reduction, Intent Consistency Constraint and Sentence Fluidity Supervision.

Since the utterance structure changes little in

IUR, one can get abundant weak label training data by deleting the informative common span between a query and its context in a large raw dialogue corpus. Pretrained from these weak data is proved efficient by Teresa(Liu et al., 2021), Few-shot generative QA query rewriting(Yu et al., 2020) and many other work. PAC(Pan et al., 2019) seeks to get additional gain in performance by imitating previous two-stage methods. The idea is to detect keywords first and then append those words to the context. However, the error accumulation is not avoided in PAC. SRL(Xu et al., 2020) train a model of semantic role labeling to highlight the core meaning of keywords in dialogue as a kind of prior knowledge for the model. Teresa uses a rank algorithm to calculate the importance of each token and pass it to following steps. (Su et al., 2019; Hao et al., 2020; Liu et al., 2020) design their models in different architectures while sharing the same advantage of reduced search space and achieve good performance. One important feature of IUR is that the rewritten query must be complete and semantically equal to previous context and incomplete query. Thus, a natural idea is to add additional task to push the model to follow this trait. CREAD(Tseng et al., 2021) adopts a binary classification task to decide whether the original query intent is complete or not. Teresa performs an KL-Divergence loss between the original and rewritten query to force their intent to be same. Lastly, although sequence-tagging model architectures are achieving best performance these days, they face the problem of readability as they do not have a large vocabulary. RUN adds additional connection words into the context. RAST uses reinforcement learning to supervise the sentence fluidity of predictions.

As mentioned before, our method fully explores two critical traits of IUR: keywords detection and intent consistency constraint. We utilize contrastive learning and multi-task learning to avoid accumulated errors in words detection. Besides, we argue that intent consistency constraint is not fully utilized yet and models can gain more from intent consistency constraint by modeling this trait in a contrastive learning way.

3 Methodology

3.1 Task Definition

Here we give the formal definition of IUR. Given the dialogue history and current utterance as (H, U_n), $H = (U_1, U_2, ..., U_n)$ is the history utterances of the dialogue. The target of IUR is to learn a function to rewrite U_n to R: $f(H, U_n) \rightarrow R$. We need to notice that R and (H, U_n) are semantically equivalent and R is self-contained, i.e. R could be understood without context.

3.2 Baseline

Our baseline is based on the current state-of-the-art model RUN (Liu et al., 2020), so we first give a brief introduction to RUN. The main modules of RUN are shown in purple cells in Figure 1. The model is defined as:

$$P_{mat} = f(CQ) \quad (1)$$

Specifically, CQ is the input that denotes concatenated history context C and current utterance Q. The model learns a mapping function f to predict from CQ to the word-level edit operation matrix $P_{mat} \in R^{M \times N}$.

The objective function is defined as:

$$L_{mat} = \frac{1}{M \times N} \sum_{i=0}^{M \times N} CE(P_{mat}^i, Y_{mat}^i) \quad (2)$$

where Y_{mat}^i is the target edit operation of pixel-level sample i. CE is the notation of cross-entropy loss.

3.3 Words Detection

Our first additional task is keywords detection. This task is inspired by two-stage methods of equipping the encoder with the capability of detecting coreferred words while not introducing accumulated error.

As shown in Figure 2, we add a **W**ords **D**etection module (WD) on top of the encoder, which performs a binary classification task for each token. WD is composed of a one-layer feed-forward neural network and takes the hidden states of joint encoded $CQ \in R^{(M+N) \times H}$ as input. The output is a binary probability distribution $P_{detect} \in R^{(M+N) \times 2}$. This is a sequence tagging problem in essence and the loss function of the word detector is as follows:

$$L_{det} = \frac{1}{M+N} \sum_{i=0}^{M+N} CE(P_{det}^i, Y_{det}^i) \quad (3)$$

Y_{det}^i is the golden keywords label of sample i. The target of word detector is to minimize the average

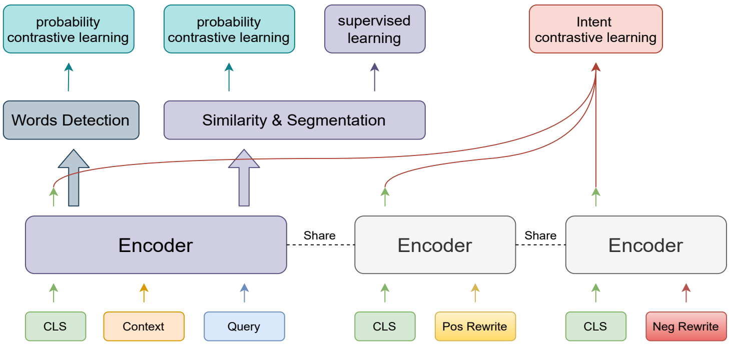
probability contrastive learning
probability contrastive learning
supervised learning
Intent contrastive learning
Words Detection
Similarity & Segmentation
Encoder
Share
Encoder
Share
Encoder
CLS
Context
Query
CLS
Pos Rewrite
CLS
Neg Rewrite

Figure 1: The framework of our proposed model. Purple cells are main modules of baseline model (RUN) and other cells are our constructed multi-tasks.

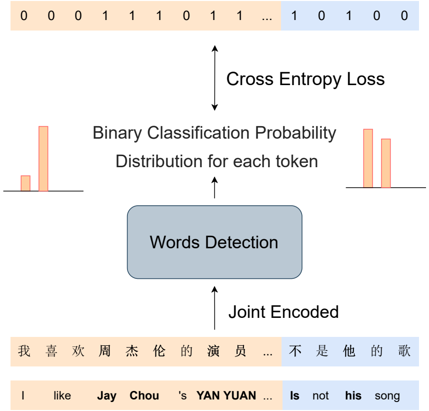
0 0 0 1 1 1 0 1 1 ... 1 0 1 0 0
Cross Entropy Loss
Binary Classification Probability Distribution for each token
Words Detection
Joint Encoded
我 喜 欢 周 杰 伦 的 演 员 ... 不 是 他 的 歌
I like Jay Chou 's YAN YUAN ... Is not his song

Figure 2: Word Detector

cross-entropy between predictions and labels. Due to the nature of modeling IUR as a sequence tagging problem (predicting edit operations for each token), we can easily assign the detection label for each token which requires no additional labeling resources. An example is shown in Figure 2, "演员"(YAN YUAN) and "周杰伦"(Jay Chou) from context are keywords as they are the omitted and the coreferred nouns. "不"(Is) and "他"(his) from current utterance are keywords as they represent the positions where omission and coreference occur.

3.4 Intent Consistency Constraint via Contrastive Learning

Intent **C**onsistency **C**onstraint (ICC)(Liu et al., 2021) is important in IUR as the rewritten query is supposed to be consistent with the contextual query in the intent space. In the meantime, contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006). We observe that the definition of ICC is naturally aligned with the purpose of contrastive learning. Thus, we introduce **C**ontrastive **L**earning based **I**ntent **C**onsistency **C**onstraint(CLICC) to the model. The objective of CLICC is to pull close the intent of joint CQ and gold rewrite R but push apart the intent of joint CQ and incomplete queries. The steps are as follows:

Anchor As shown in Figure 1, we insert a "[CLS]" token before the concatenated context and query. The hidden states of "[CLS]" token represents the intent of joint CQ.

Positive Instance According to the prior assumption of IUR, a gold rewrite R is supposed to be consistent with the contextual query in intent dimension, so we use R as positive instance. We first insert a "[CLS]" token before the gold rewrite R and then feed R into the same encoder to get the encoded hidden states of "[CLS]" in R.

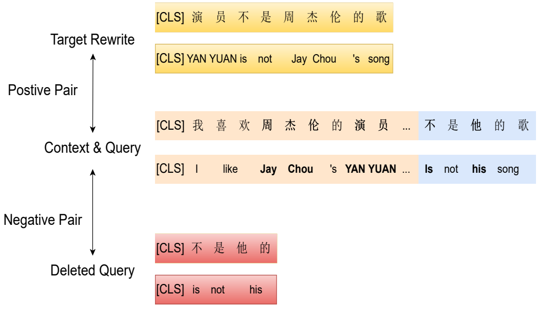
Target Rewrite
[CLS] 演 员 不 是 周 杰 伦 的 歌
[CLS] YAN YUAN is not Jay Chou 's song
Postive Pair
Context & Query
[CLS] 我 喜 欢 周 杰 伦 的 演 员 ... 不 是 他 的 歌
[CLS] I like Jay Chou 's YAN YUAN ... Is not his song
Negative Pair
Deleted Query
[CLS] 不 是 他 的
[CLS] is not his

Figure 3: An example of query needs rewriting. When the intent of original query is not complete, the deletion could further corrupt the query into a more incomplete one.

Hard Negative Instances For the construction of hard negative instances, we adopt a simple yet effective strategy: Random Token Deletion. We randomly delete 20% tokens in the original query for each sample. Then we insert a "[CLS]" token before the deleted query and feed it through the

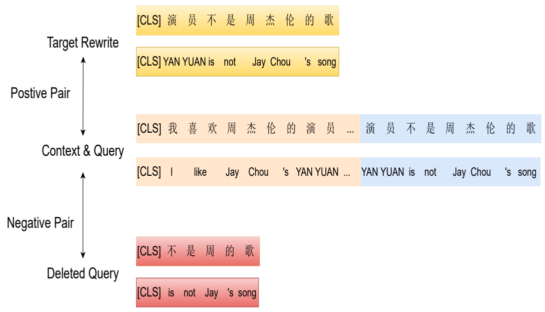
[CLS] 演 员 不 是 周 杰 伦 的 歌
Target Rewrite
[CLS] YAN YUAN is not Jay Chou 's song
Postive Pair
[CLS] 我 喜 欢 周 杰 伦 的 演 员 ... 演 员 不 是 周 杰 伦 的 歌
Context & Query
[CLS] I like Jay Chou 's YAN YUAN ... YAN YUAN is not Jay Chou 's song
Negative Pair
[CLS] 不 是 周 的 歌
Deleted Query
[CLS] is not Jay 's song

Figure 4: An example of query does not need rewriting. When the intent of original query is complete, the deletion could corrupt the query into an incomplete one.

same encoder to get the intent representation of the deleted query. Because of similar sentence structure, these instances are hard to be distinguished from the original query. This strategy is effective in three ways:

(a) Token deletion ensures a complex negative instance for each sample even if the original queries are complete.

(b) Token deletion for incomplete original queries lowers the learning difficulty for the model in beginning epochs, as this strategy creates a more severely incomplete query, which is farther from the complete query.

(c) Random token deletion enriches the negative instances as the deleted queries for a certain sample i are different between epochs.

As shown in Figure 3, the original query is incomplete and random token deletion further corrupts the query. As shown in Figure 4, random token deletion converts the complete query into an incomplete one.

Easy Negative Instances We also fill the negative sample space with abundant, accessible negative instances. For a certain sample i in one batch, other positive and hard negative instances will be treated as negative pairs for sample i, as their intent is naturally different.

Loss function Following (Chen et al., 2020), we adopt the normalized temperature-scaled cross-entropy loss (NT-Xent) as the contrastive objective for CLICC. Suppose we have N randomly sampled dialogues from the training set as a mini-batch during each training step. There are N representations for all positive instances, negative instances and anchors in the batch. The objective function is slightly modified to train each anchor to find its counterpart among $3N - 2$ in-batch negative samples:

$$L_{icon} = -log \frac{exp(sim(r_i, r_i^+)/\tau)}{Anchor + Pos + Neg} \quad (4)$$

$$Anchor = \sum_{j=1}^{N} \mathbb{I}_{[j \neq i]} exp(sim(r_i, r_j)/\tau) \quad (5)$$

$$Pos = \sum_{j=1}^{N} \mathbb{I}_{[j \neq i]} exp(sim(r_i, r_j^+)/\tau) \quad (6)$$

$$Neg = \sum_{j=1}^{N} exp(sim(r_i, r_j^-)/\tau) \quad (7)$$

where sim(·) is defined as the cosine similarity function, τ in temperature parameter and $\mathbb{I}$ is the indicator function. *icon* is the short notation of **I**ntent **Con**trastive.

3.5 Probability Space Contrastive Learning

Inspired by the effective result of CLICC, we introduce **P**robability **C**ontrastive **L**earning (**PCL**) modules on top of both words detection and semantic segmentation modules to help the optimization in learning. This target target is to make the predicted probability distribution closer for a positive pair. The steps are as follows:

Positive Instances One way to effectively create positive instances in NLP tasks is through data augmentation such as word reordering, deletion, repeating and substitution(Feng et al., 2021). However, these augmentation are not suitable for the PCL module in IUR in two aspects: a) we may unconsciously delete or repeat the keywords in context and queries. b) We could potentially change the original intent of dialogues. These two risks both lead to a false predicted distribution. Thus, instead of above data augmentation techniques, similar to ideas in SimCSE (Gao et al., 2021), we use Dropout(Srivastava et al., 2014) to safely acquire the positive instances. Specifically, we feed the same dialogue utterances to the encoder and embedder twice to get two different dropped context and query CQ_1,CQ_2. We use CQ_1 as the anchor and CQ_2 and the positive instance.

Loss function As shown in Figure 5 and Figure 6, we constrain the predicted distributions of WD and word-level edit matrix of a positive pair to be close by minimizing the bidirectional Kullback-Leibler(Bi-KL) divergence loss between predicted distributions as follows:

$$L_{pcon} = \frac{1}{2}(L_{det} + L_{mat}) \tag{8}$$

where $pcon$ is notation of **p**robability, L_{det} and L_{mat} denote Bi-KL divergence loss of words detection and edit matrix respectively:

$$L_{det} = KL(P_{det}^{CQ_1}||P_{det}^{CQ_2}) + KL(P_{det}^{CQ_2}||P_{det}^{CQ_1}) \tag{9}$$

$$L_{mat} = KL(P_{mat}^{CQ_1}||P_{mat}^{CQ_2}) + KL(P_{mat}^{CQ_2}||P_{mat}^{CQ_1}) \tag{10}$$

where det and mat are the notation of **det**ection and **mat**rix respectively. KL is Bi-KL divergence loss.

Negative Instances We do not need negative instances in PCL as we are not being exposed to the risk of model collapse compared to representation contrastive learning. Model collapse is the extreme opposite of Uniformity explained in (Wang and Isola, 2020), which means all representations of data are centering to one point in the hypersphere. This is avoided in PCL because though bidirectional KL is pulling two distributions closer because the original supervised target ensures the correct optimization direction.

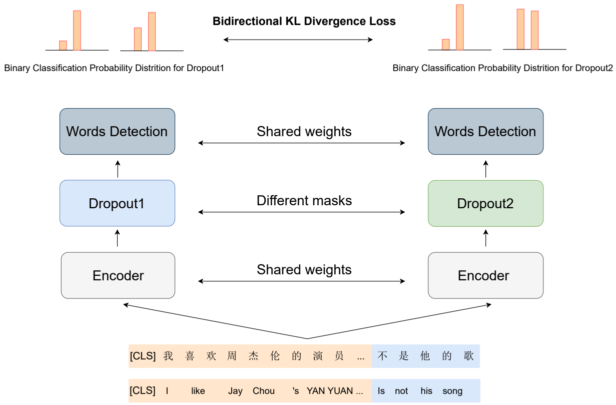
Bidirectional KL Divergence Loss
Binary Classification Probability Distrition for Dropout1
Binary Classification Probability Distrition for Dropout2
Words Detection
Shared weights
Words Detection
Dropout1
Different masks
Dropout2
Encoder
Shared weights
Encoder
[CLS] 我 喜 欢 周 杰 伦 的 演 员 ... 不 是 他 的 歌
[CLS] I like Jay Chou 's YAN YUAN ... Is not his song

Figure 5: PCL on word detection probability distribution.

3.6 Final Learning objectives

Finally, we combine all tasks together and train them simultaneously by taking the weighted summation of all loss functions and the final loss function is shown as Equation 11

$$\begin{aligned} L_{forward} &= L_{mat}^{CQ_1} + L_{mat}^{CQ_2} + \alpha(L_{det}^{CQ_1} + L_{det}^{CQ_2}) \\ L_{final} &= L_{forward} + \beta(L_{icon}) + \gamma(L_{pcon}) \end{aligned} \tag{11}$$

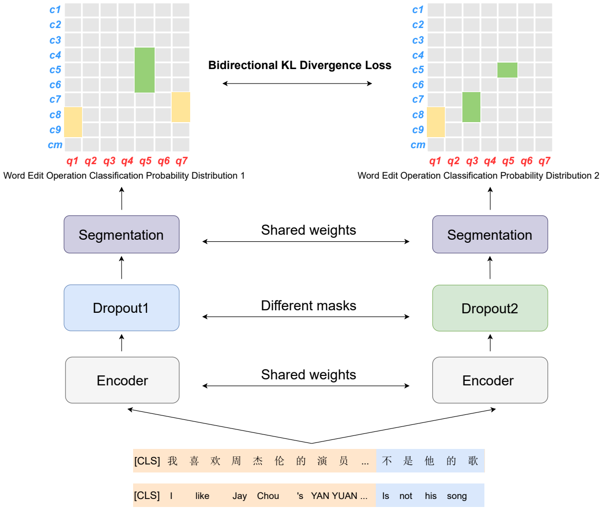
c1
c2
c3
c4
c5
c6
c7
c8
c9
cm
q1 q2 q3 q4 q5 q6 q7
Word Edit Operation Classification Probability Distribution 1
Bidirectional KL Divergence Loss
c1
c2
c3
c4
c5
c6
c7
c8
c9
cm
q1 q2 q3 q4 q5 q6 q7
Word Edit Operation Classification Probability Distribution 2
Segmentation
Shared weights
Segmentation
Dropout1
Different masks
Dropout2
Encoder
Shared weights
Encoder
[CLS] 我 喜 欢 周 杰 伦 的 演 员 ... 不 是 他 的 歌
[CLS] I like Jay Chou 's YAN YUAN ... Is not his song

Figure 6: PCL on word edit operation probability distribution

where α, β, γ are coefficients for three introduced addition learning tasks, WD, CLICC and PCL.

4 Experiments

In this section, we conduct thorough experiments to demonstrate the effectiveness of our approach.

4.1 Datasets

We conduct experiments on two Chinese public datasets in open-domain dialogues: MULTI (Pan et al., 2019) and REWRITE (Su et al., 2019). We use the same data split method for these datasets as their original paper. We display the statistics of two datasets in Table 2.

	MULTI	REWRITE
Train	$194K$	$18K$
Dev	$5K$	$2K$
Test	$5K$	N/A
Avg. C len	25.5	17.7
Avg. Q len	8.6	6.5
Avg. R len	12.4	10.5

Table 2: Statistics of the datasets. NA means the we use development set as test set. "Avg" is short for average, "C" for context, "Q" for current query, "R" for rewritten query.

4.2 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., 2020), PAC(Pan et al., 2019), RAST(Hao et al., 2020), T-Ptr-λ (Su et al., 2019) and RUN(Liu et al., 2020).

4.3 Evaluation Metrics

We take automatic metrics to evaluate our approach. Following (Pan et al., 2019), we employ the widely used automatic metrics BLEU(Papineni et al., 2002), ROUGE(Lin, 2004), Exact-Match(EM) and Rewriting F-score(Pan et al., 2019). (i) **BLEU**$_n$ (**B**$_n$) evaluates how similar the rewritten utterances are to the golden ones via the cumulative n-gram BLEU score. (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. (iii) **EM** stands for the exact match accuracy, which is the strictest evaluation metric. (iv) Rewriting **Precision**$_n$, **Recall**$_n$ and **F-score**$_n$ (P_n, R_n, F_n) emphasize more on how well we recover the correferred words.

4.4 Implementation Details

Our approach is developed based on the model architecture of RUN. We follow the original settings in RUN: The weighted cross-entropy loss is used; We used Adam(Kingma and Ba, 2017) to optimize the model and set the learning rate 1e-3, except for BERT(Devlin et al., 2019) as 1e-5; The embedding size and hidden size are 200 respectively. Specifically, BERT aforementioned is $BERT_{base}$.

4.5 Results

Our results on MULTI(Pan et al., 2019) and REWRITE (Su et al., 2019) are shown in Table 3 and Table 4 respectively. On both datasets, our model equipped with one extra module (WD) already surpasses the existing best model on all metrics. The full model equipped with all three modules largely improve the overall performance on each metric.

Observing from the ablation results of three modules, the F-score performance would drop a bit while BLEU and ROUGE are better after adding CLICC. Through manually analyzing the predicted utterance before and after adding CLICC, we find that CLICC helps reduce the repetition of keywords in generated utterances. This could improve the fluidity and correctness of the sentence while restraining the ability of WD module, which causes the drop of F-score. However, we argue that this feature is useful as we can adjust the importance of two modules for different tasks. For example, the downstream tasks such as FAQ in the dialogue system pay more attention to the recovered keywords would benefit more from WD. In the meantime, if we would show the rewritten queries to users, a more fluent and correct utterance may fit better. Finally, adding PCL boosts the performance again.

4.6 Influence of Temperature

The temperature τ in NT-Xent loss Equation 4 is used to control the smoothness of the distribution normalized by softmax operation. A large temperature smooths the distribution while a small temperature sharpens the distribution. A smoother distribution is easier to learn while risking being not discriminative enough. We explore the influence of temperature in Figure 7. The performance is sensitive to the temperature. A unsuitable temperature will degrade the model performance. The optimal temperature is obtained around 0.5. This phenomenon demonstrates that, as most negative sentences are far to each other (naturally semantically different), a small temperature may make this task too hard to learn since the model should learn a more general difference rather than detailed differences between anchor and negative samples. A too large temperature is also inappropriate as it may hide the general differences among samples.

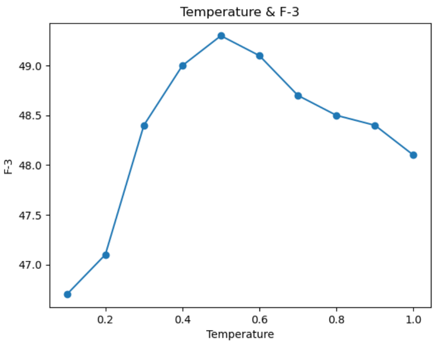

Figure 7: The relation between temperature and F3. We get best F3 when temperature is 0.5.

4.7 Different negative strategies

How to construct hard semantically negative samples is critical for CLICC. One can naturally come up with two ideas: a) Since anchor is semantically complete, we can use original incomplete query as the negative sample. b) Erasing the longest informative common span between context and current query, which could corrupt the query into a incom-

Model	F_1	F_2	F_3	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{R}_1$	$\mathbf{R}_2$
SRL	NA	NA	NA	85.8	82.9	89.6	83.1
T-Ptr-λ (n_beam=5)	51.0	40.4	33.3	90.3	87.7	90.1	83.0
PAC(n_beam=5)	63.7	49.7	40.4	89.9	86.3	91.6	82.8
SARG(n_beam=5)	62.3	52.5	46.4	91.4	88.9	91.9	85.7
RAST	NA	NA	NA	89.7	88.9	90.9	84.0
RUN	69.0	57.1	48.8	90.7	87.7	92.0	85.1
+WD	70.8	58.2	49.6	91.1	88.1	92.1	85.2
+CLICC	70.2	57.8	49.3	91.5	88.6	92.3	85.7
+PCL	**71.1**	**59.1**	**51.1**	**92.1**	**89.4**	**92.6**	**86.2**

Table 3: Reuslts on Restoration-200k. All models except T-Ptr-λ are initalized from pretrained Bert-base-Chinese model. All results are extracted from the original papers except RUN. For RUN, we reproduce the results from released code to ensure a fair comparison as we are adding modules onr RUN. The final line is the result of our complete model equipped with all three modules.

Model	$F1$	$F2$	$F3$	**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
+WD	**90.5**	82.8	77.2	68.1	**94.5**	**92.0**	86.9	**95.8**	90.9	94.5
+CLICC	90.1	82.7	77.3	68.2	94.1	91.7	86.8	95.7	90.8	94.5
+PCL	89.8	**83.2**	**78.2**	**69.0**	93.7	91.5	**87.0**	95.6	**91.0**	**94.6**

Table 4: Results on Rewrite-20k. All models are initialized from pretrained Bert-base-Chinese model. All results are extracted from the original papers except RUN. For RUN, we reproduce the results from released code to ensure a fair comparison as we are adding modules on RUN. The final line is the result of our complete model equipped with all three modules.

plete or more incomplete one. These two strategies are natural as they follow the intention of CLICC which is to force the intent consistency. However, a common disadvantage of these two methods is that they are consistent through the training, i.e. the content of hard semantically negative samples for a certain data i would not change in the training process. This results in a relatively stagnation in local optimal for the representation of intent. Thus, we adopt the random deletion as our final strategy. The comparison of performance on Rewrite dataset is shown in Table 5.

Strategies	F_3	**EM**	$\mathbf{B}_4$	$\mathbf{R}_L$
ORIGIN	76.7	67.0	86.8	94.3
SE	76.4	68.0	86.7	94.0
RD	77.3	69.0	87.0	94.6

Table 5: Reuslts on Rewrite-20k with different negative strategies. ORIGIN means we use original incomplete query as negative samples. SE represents Span Erase, which means we erase the commmon span bettwen context and current query. RD means random deletion which is explained in section about CLICC.

5 Conclusion

In this work, we explore the key traits of utterance rewriting. We adopt contrastive learning method to model the intent consistency at sentence level and probability consistence in probability space. With the help of carefully designed combination of multi-tasks, our approach achieves the best performance on several public datasets. In the future, we will explore modeling more effective positive and negative samples in contrastive learning to improve the utterance rewriting.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2020. Robust dialogue utterance rewriting as sequence tagging. *arXiv preprint arXiv:2012.14535*.

Mengzuo Huang, Feng Li, Wuhe Zou, and Weidong Zhang. 2020. Sarg: A novel semi autoregressive generator for multi-turn incomplete utterance restoration.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Vineet Kumar and Sachindra Joshi. 2016. Non-sentential question resolution using sequence to sequence learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2022–2031, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Hang Liu, Meng Chen, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Conversational query rewriting with self-supervised learning. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7628–7632.

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.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

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*, 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.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic role labeling guided multi-turn dialogue rewriter. *arXiv preprint arXiv:2010.01417*.

Qingyu Yin, Weinan Zhang, Yu Zhang, and Ting Liu. 2017. A deep neural network for chinese zero pronoun resolution.

Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. 2018. Zero pronoun resolution with attention-based neural network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 13–23, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. *Few-Shot Generative Conversational Query Rewriting*, page 1933–1936. Association for Computing Machinery, New York, NY, USA.

Zhuosheng Zhang and Hai Zhao. 2021. Advances in multi-turn dialogue comprehension: A survey. *arXiv preprint arXiv:2103.03125*.

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.