How Well Apply Simple MLP to Incomplete Utterance Rewriting?

Jiang Li[1,2] **, Xiangdong Su**[1,2] *, **Xinlan Ma**[1,2] **, Guanglai Gao**[1,2]
[1] College of Computer Science, Inner Mongolia University, Hohhot, China
[2] National & Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot, China
lijiangimu@gmail.com, cssxd@imu.edu.cn, xinlanm0127@163.com, csggl@imu.edu.cn

Abstract

Incomplete utterance rewriting (IUR) aims to restore the incomplete utterance with sufficient context information for comprehension. This paper introduces a simple yet efficient IUR method. Different from prior studies, we first employ only one-layer **M**LP architecture to mine latent semantic information between joint utterances for **IUR** task (**MIUR**). After that, we conduct a joint feature matrix to predict the token type and thus restore the incomplete utterance. The well-designed network and simple architecture make our method significantly superior to existing methods in terms of quality and inference speed[1].

Turn	**Utterance** (Translation)
u_1	你知道新冠肺炎吗 Do you know COVID-19
u_2	是的，我知道 Yes, I know
u_3	那是什么 What is that
u_3'	新冠肺炎是什么 What is COVID-19

Table 1: An example of incomplete utterance rewriting. u_1 and u_2 denote the context utterances. u_3 is the incomplete utterance. u_3' is the rewritten utterance.

1 Introduction

Multi-turn dialogue modeling is a research area focusing on developing systems that can engage in multiple conversation turns with humans. This type of modeling is often used in the field of human-machine interaction to improve the ability of artificial intelligence systems to communicate with humans in a natural and intuitive way. One of the challenges of multi-turn dialogue modeling is to accurately understand and respond to the context and meaning of the conversation, as well as to handle incomplete or ambiguous utterances that may be used for brevity or to convey meaning. As shown in Table 1, the incomplete utterance u_3 refers to the semantic of "新冠肺炎" (COVID-19) with "那" (that). The limited context provided by a single utterance, such as u_3, can lead to referential ambiguity and semantic incompleteness in downstream applications like retrieval-based dialogue systems, as demonstrated in a study by Ni et al. (2022). In addition, Su et al. (2019) has revealed that coreference and ellipsis are prevalent in more than 70% of utterances, particularly in pro-drop languages like Chinese. These linguistic phenomena in conversation present a significant challenge for the development of practical conversational AI systems.

To address this issue, recent works (Kumar and Joshi, 2016; Su et al., 2019; Pan et al., 2019; Xu et al., 2020) proposed the Incomplete Utterance Rewriting (IUR) task, which aims to transform an incomplete or context-dependent statement into a self-contained, semantically equivalent one that can be understood without any additional context. As shown in Table 1, IUR ($u_3 \rightarrow u_3'$) task makes the downstream dialogue modeling more precise.

Despite previous works achieving promising results, the speed of autoregressive generation remains a limiting factor. To improve the speed, Huang et al. (2021) fuses the sequence labeling and non-autoregressive generation, which predicts missing elements in incomplete utterance and rewritten utterance. In addition, Liu et al. (2020) formulates IUR as semantic segmentation task based on U-Net (Ronneberger et al., 2015) and achieves better performance at a faster speed. However, above mentioned models are still not simple enough.

In this paper, we propose a simple yet efficient

*Corresponding author
[1]Our code is available at https://github.com/IMU-MachineLearningSXD/MIUR

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1567–1576

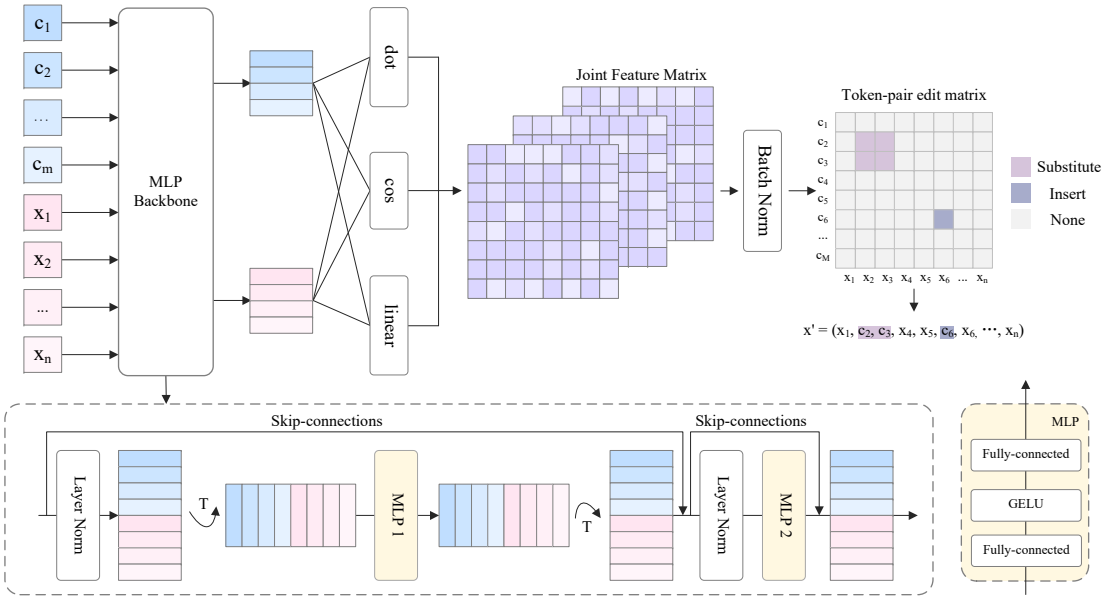


Figure 1: The architecture of our proposed model.

solution that our model first employs MLP architecture to simultaneously mine the semantic associations between the context utterances and the incomplete utterance, and capture attention information between them. After MLP architecture, we obtain the joint feature maps and further construct the token-pair edit matrix. Finally, the above matrix is edited according to prediction edit type tokens to generate the final rewritten utterance. Experiments show that our approach achieves better performance on several datasets across different domains and languages with low resource costs and a much faster inference speed.

2 Methodology

In this section, we elaborate on our proposed approach. As shown in Figure 1, our method mainly consists of two modules: MLP backbone network and joint feature matrix. For a multi-turn dialogue utterances $(u_1, u_2, ..., u_t)$, we concatenate all the context utterances to produce an m-length word sequence $c = (c_1, c_2, ..., c_m)$ and employ a special mask $[SEP]$ to separate different context utterances. Meanwhile, all the incomplete utterances are denoted as an n-length word sequence $x = (x_1, x_2, ..., x_n)$.

2.1 MLP Backbone Network

We first concatenate the context utterances and the incomplete utterances to construct a joint $m + n$ length word sequence $\mathcal{H} = (c_1, c_2, ..., c_m, x_1, x_2, ..., x_n)$. Besides, pretrained language models have been found to be highly effective in various natural language processing tasks. Hence, we employ BERT (Devlin et al., 2019) to initialize the word vector matrix $\mathbf{H}$, where $\mathbf{H} \in \mathbb{R}^{(m+n)\times 768}$. MLP backbone network contains two MLP blocks. Specifically, the first MLP block is responsible for mining the global semantic association information between context utterances c and incomplete utterance x. The second MLP block aims to learn the confidence level for each word embedding. This further enables the model to focus on important word information. It is important for the follow-up edit type classification, including substitute, insert and none. Each MLP block contains two fully-connected layers and a nonlinearity applied independently. For clarity and simplicity, we exclude the transposition process and the whole process can be represented as:

$$\begin{aligned} \mathbf{I}_{*,i} &= \mathbf{H}_{*,i} + \mathbf{W_2}\sigma(\mathbf{W_1} LN(\mathbf{H}_{*,i})), \\ \mathbf{K}_{j,*} &= \mathbf{I}_{j,*} + \mathbf{W_4}\sigma(\mathbf{W_3} LN(\mathbf{I}_{j,*})), \end{aligned} \tag{1}$$

where $i = 1, 2, .., 768$, $j = 1, 2, .., m + n$ and σ represents GELU (Hendrycks and Gimpel, 2016). In addition, MLP backbone contains other standard architectural components: skip-connections (He et al., 2016) and LayerNorm (LN) (Ba et al., 2016).

In contrast to the approach taken by Tolstikhin et al. (2021), who treated the word vector matrix $\mathbf{H}$ as an image and employed 1×1 convolution on

non-overlapping image patches, we directly input the word vector matrix $\mathbf{H}$ into the MLP backbone network. Our operation avoids the loss of semantic spatial information resulting from 1×1 convolution. Furthermore, since the number of words in each utterance varies, we utilize padding operation and copy mechanism (Gu et al., 2016; Zeng et al., 2018) to maintain a consistent sequence length. It is worth noting that our approach employs a one-layer MLP backbone network.

2.2 Joint Feature Matrix

Furthermore, to further capture the relevance between word embeddings, we employ three similarity functions: dot product similarity (*dot* Sim.), cosine similarity (*cos* Sim.), and linear similarity (*linear* Sim.). The word-to-word embeddings relevance between each context utterance's word embedding $\mathbf{K}_{c_m}$ and each incomplete utterance's word embedding $\mathbf{K}_{x_n}$ are captured using a 3-dimensional joint feature matrix $\mathbf{J}(c_m, x_n)$ represented as follows:

$$\mathbf{J}(c_m, x_n) = [\mathbf{K}_{c_m} \cdot \mathbf{K}_{x_n}; \cos(\mathbf{K}_{c_m}, \mathbf{K}_{x_n}); linear(\mathbf{K}_{c_m}, \mathbf{K}_{x_n})]. \quad (2)$$

Finally, we employ BatchNorm (Ioffe and Szegedy, 2015) on joint feature matrix $\mathbf{J}(c_m, x_n)$ to expedite and stabilize the training process. The batch is obtained by computing the mean and variance of the batch activation, which captures global information. After applying the BatchNorm operation, the matrix $\mathbf{J}(c_m, x_n)$ is flattened, and each feature vector is mapped to one of three token types: Substitute, Insert, or None. This generates the token-pair edit matrix.

2.3 Supervised Label

Prior to training our model in the supervised fashion, we need to create word-level labels through the following process to construct our training set. Specifically, we first calculate the longest common subsequence (LCS) between the incomplete utterance and the rewritten utterance. Then, we align the incomplete utterance, the rewritten utterance, and the LCS using a greedy strategy. Finally, we identify the corresponding tokens in the rewritten utterance and mark them accordingly. Please refer to Algorithm 1 in Appendix A for a detailed description.

3 Experiments

3.1 Experimental Setup

Datasets We conduct the experiments on three IUR benchmarks from different domains and languages, including RESTORATION-200K (Pan et al., 2019), REWRITE (Su et al., 2019) and CANARD (Elgohary et al., 2019). The statistics of the datasets are shown in Appendix B.

Baselines We compare the performance of our method with the following baselines: (i) **Generation models** need to generate rewritten utterances from scratch, including Seq2Seq model L-Gen (Bahdanau et al., 2015), the hybrid pointer generator network L-Ptr-Gen (See et al., 2017), the basic transformer models T-Gen and T-Ptr-Gen (Vaswani et al., 2017), Syntactic (Kumar and Joshi, 2016), PAC (Pan et al., 2019), L-Ptr-λ and T-Ptr-λ (Su et al., 2019). The above models are limited by the speed of generation. (ii) **Structure aware models** contain RUN(Liu et al., 2020) and SARG (Huang et al., 2021).

For more information about other experimental setups, please see Appendix B.

3.2 Main Results

Table 2 shows the experimental results on RESTORATION-200K. Our proposed approach, MIUR, achieves competitive results compared to all previous State-of-the-Art methods as shown in Table 2. The results indicate MIUR can effectively mine the semantic information between utterances with two types of MLP architecture. Furthermore, we discovered that MIUR places more emphasis on rewriting precision ($\mathcal{P}_n$) metrics. The first MLP architecture captures global semantic associations between context utterances and incomplete utterance, while the second MLP architecture focuses more on significant word embedding information. Our approach effectively combines two different MLPs and provides an effective guideline for the subsequent construction of the joint feature map matrix, leading our approach to concentrate more on essential word information and to pursue higher rewriting precision. Additionally, we achieve comparable Recall$_n$ results to the baselines. The experimental results of REWRITE and CANARD also come to the same conclusion, which can be found in Appendix C.

Model	$\mathcal{P}_1$	$\mathcal{R}_1$	$\mathcal{F}_1$	$\mathcal{P}_2$	$\mathcal{R}_2$	$\mathcal{F}_2$	$\mathcal{P}_3$	$\mathcal{R}_3$	$\mathcal{F}_3$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{R}_1$	$\mathbf{R}_2$
Syntactic	67.4	37.2	47.9	53.9	30.3	38.8	45.3	25.3	32.5	84.1	81.2	89.3	80.6
L-Gen	65.5	40.8	50.3	52.2	32.6	40.1	43.6	27.0	33.4	84.9	81.7	88.8	80.3
L-Ptr-Gen	66.6	40.4	50.3	54.0	33.1	41.1	45.9	28.1	34.9	84.7	81.7	89.0	80.9
PAC	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
T-Ptr-$\lambda^{\heartsuit}$	-	-	51.0	-	-	40.4	-	-	33.3	90.3	87.4	90.1	83.0
SARG$^{\heartsuit}$	-	-	62.4		-	52.5	-	-	46.3	92.2	89.6	92.1	**86.0**
RUN	73.2	**64.6**	68.6	59.5	**53.0**	56.0	50.7	45.1	47.7	92.3	89.6	92.4	85.1
MIUR (Ours)	**76.4**	63.7	**69.5**	**62.7**	52.7	**57.3**	**54.3**	**45.9**	**49.7**	**93.0**	**90.1**	**92.6**	85.7

Table 2: Experimental results on RESTORATION-200K. All results are taken from the original papers. Dashes: results are not reported in the responding literature. $\heartsuit$: results are derived from (Huang et al., 2021).

3.3 Inference Speed

Table 3 presents a comparison of the inferential speed of our model with the baselines. All models were implemented in PyTorch and run on a single NVIDIA V100. We can observe that the proposed MIUR achieves the fastest inference speed compared with the SOTA methods. Specifically, MIUR's speed is 3.14 times faster than that of L-Gen (n_Beam=1). Moreover, Compared with RUN in the second place, MIUR achieves 20% improvement in the inference speed. This enhanced performance can be attributed to the fact that our model employs only a one-layered MLP backbone to capture inter-utterances semantic information, without utilizing other modules. The simplified architecture, thus, contributes to the model's faster inference speed without compromising the performance.

Model	Speedup
L-Gen (n_Beam=1)	1.00 ×
L-Ptr-Net (n_Beam=1)	0.57 ×
L-Ptr-Gen (n_Beam=1)	0.93 ×
T-Gen (n_Beam=1)	0.25 ×
T-Ptr-Net (n_Beam=1)	0.13 ×
T-Ptr-Gen (n_Beam=1)	0.14 ×
SARG (n_Beam=1)	2.63 ×
RUN	2.61 ×
MIUR (Ours)	**3.14** ×

Table 3: The inference speed comparison between MIUR and baselines on RESTORATION-200K. n_Beam stands for the beam size in beam search, not applicable for RUN and MIUR.

3.4 Ablation Study

To verify the effectiveness of MLP architecture in our model, we conduct a thorough ablation study in Table 4. Notably, EM and $\mathcal{P}_2$ metrics significantly decreased when the model did not use MLP backbone architecture. The results again prove that MLP backbone can effectively mine latent semantic information between utterances and provide more precise guidance for the follow-up edit type classification. In addition, MIUR uses only one type of MLP architecture alone can also lead to performance degradation. Since the first MLP architecture can effectively mine the semantic associations between context utterances and incomplete utterance, and the second MLP architecture increased focus on capturing attention information between utterances. It's only with the full MLP structure that MIUR can capture semantic information more accurately and to a wider extent.

w/o MLP	MLP 1	MLP 2	EM	$\mathcal{P}_2$	$\mathcal{R}_2$	$\mathcal{F}_2$	$\mathbf{B}_2$
	✔	✔	**67.7**	**86.1**	**78.6**	**82.2**	**91.2**
	✔		66.4	84.8	78.3	81.4	90.6
		✔	66.6	85.4	78.1	81.6	90.7
✔			65.1	82.4	77.3	80.1	90.5

Table 4: The ablation results on REWRITE dataset.

As mentioned in Section 2.1, we perform an ablation study about using two different padding strategies to ensure consistent sequence length. Table 5 indicates that the model obtains a small performance improvement using copy mechanism, which further increases the semantic interaction between utterances. But this operation limits inference speed. Given a tiny improvement using copy mechanism, our model employs zero padding method.

Padding Strategy	EM	$\mathcal{P}_2$	$\mathcal{R}_2$	$\mathcal{F}_2$	Speedup
zero padding	67.73	86.12	78.63	82.21	**1.00** ×
copy mechanism	67.81	86.22	78.69	82.33	0.96 ×

Table 5: The ablation results on REWRITE dataset.

3.5 More Discussion for MLP

To further investigate whether our proposed MLP backbone can effectively mine the semantic associations between utterances, we visualize the word embeddings composed of the context utterances and the incomplete utterance in Figure 2. The y-axis represents our selection of 40 words consisting of the context utterances and the incomplete utterance. The x-axis represents the features of the first 100 dimensions of our intercepted word embeddings. It is not difficult to notice that word embeddings appear more distinctly characterized by vertical stripes after MLP backbone. Consequently, this further indicates that semantic information between words is more closely related, and our method can effectively learn the semantic relatedness between words after passing through the MLP network we designed.

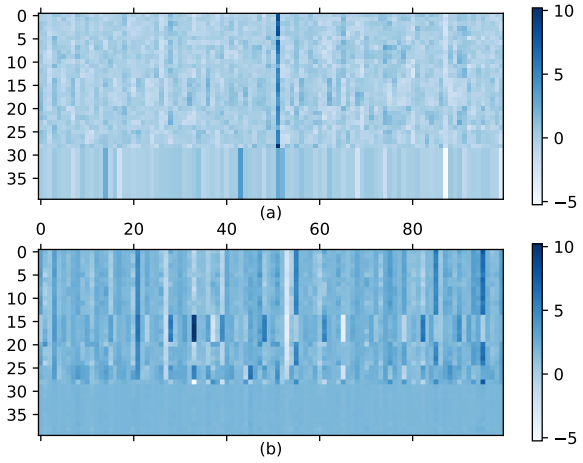


Figure 2: Visualization of word embedding matrix. (a) denotes the initial word embedding and (b) denotes the word embedding after MLP backbone.

4 Conclusion & Future Work

In this paper, we propose a simple yet effective IUR method. We utilize one-layer MLP structure to mine the inter-utterance semantic information from different perspectives. This improves the ability to predict the correct token between incomplete utterance and rewritten utterance. Benefiting from the fact that our model effectively employs MLP to IUR task, allowing our approach to achieve significant results in terms of performance and inference speed. This study represents the first preliminary exploration of the use of MLP on IUR task. In the future, we will investigate on extending our approach to other dialogue areas.

Limitations

One limitation of current token-pair edit matrix based incomplete utterance rewriting models is that they are only able to select tokens that have appeared in the context utterances. Thus, these models, including our own, are unable to generate new words, such as conjunctions and prepositions, to improve metrics such as fluency. However, this can be addressed by incorporating an additional word dictionary as proposed by Liu et al. (2020) to improve fluency for out-of-vocabulary words (OOV). In addition, we will consider combining generative models (GPT (Radford et al., 2019), T5 (Raffel et al., 2020) etc.) to assist in the recovery of the incomplete utterances in the future works.

Acknowledgement

This work was funded by National Natural Science Foundation of China (Grant No. 61762069), Key Technology Research Program of Inner Mongolia Autonomous Region (Grant No. 2021GG0165), Key R&D and Achievement Transformation Program of Inner Mongolia Autonomous Region (Grant No. 2022YFHH0077), The Central Government Fund for Promoting Local Scientific and Technological Development (Grant No. 2022ZY0198), Big Data Lab of Inner Mongolia Discipline Inspection and Supervision Committee (Grant No. 21500-5206043).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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. 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 OK 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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Mengzuo Huang, Feng Li, Wuhe Zou, and Weidong Zhang. 2021. Sarg: A novel semi autoregressive generator for multi-turn incomplete utterance restoration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13055–13063.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

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.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2022. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, pages 1–101.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas 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.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514.

A Constructing Supervised Labels

We describe here the algorithm for building word-level supervised labels. Taking Table 1 as an example, given U as "那是什么" (What is that) and U' as "新冠肺炎是什么" (What is COVID-19). Their longest common subsequence (LCS) is "是什么" (What is). Hence, "那" (that) is marked as $[DEL]$ in U and "新冠肺炎" (COVID-19) is marked as $[ADD]$ in U'. Correspondingly, the edit type (supervised label) is $Substitute$.

Algorithm 1: Construct Supervised Labels

Input: U: the incomplete utterance
U': the rewritten utterance
Output: L: the supervised label

Computing the LCS between U and U'.
for $w_x \in U$ **do**
  if $w_x \notin LCS$ **then**
    $mark(w_x) = [DEL]$
  end
end
for $w'_x \in U'$ **do**
  if $w'_x \notin LCS$ **then**
    $mark(w'_x) = [ADD]$
  end
end
The same mark is combined into one span.
Comparing U and U' at the span level.
for $(s_x, s'_x) \in (U, U')$ **do**
  if $s_x = [DEL]$ *and* $s'_x = [ADD]$ **then**
    $L = Substitute$
  else
    $L = Insert$
  end
end
return L

B Other Experimental Setups

Evaluation Following the previous works, we apply BLEU$_n$ ($\mathbf{B_n}$) (Papineni et al., 2002), ROUGE$_n$ ($\mathbf{R_n}$) (Lin, 2004), EM (exact match), Rewriting Precision$_n$, Recall$_n$ and F-score$_n$ ($\mathcal{P}_n, \mathcal{R}_n, \mathcal{F}_n$) (Pan et al., 2019) as the automatic evaluation metrics.

Implementation Details We implement our proposed model via pytorch . All experiments are trained on a single NVIDIA Tesla V100. We use Adam (Kingma and Ba, 2015) optimizer and employ grid search to find the best hyperparameters based on the performance on the validation datasets. The learning rate is set to $1e-5$ for all datasets. The best models are selected by early stopping on the validation datasets, and the max epoch is 100.

C Additional Experimental Results

Table 7 and Table 8 show the experimental results on REWRITE and CANARD, respectively. Our method also achieves competitive results on all scores. The results again demonstrate the effectiveness of our model.

	RESTORATION-200K	REWRITE	CANARD
Language	Chinese	Chinese	English
# Train	194K	18K	32K
# Dev	5K	2K	4K
# Test	5K	-	6K
Avg. Con length	25.8	17.7	85.4
Avg. Inc length	8.6	6.5	7.5
Avg. Rew length	12.4	10.5	11.6

Table 6: Statistics of three experimented datasets. "Avg" for average, "Con" for context utterance, "Inv" for incomplete utterance, "Rew" for rewritten utterance.

Model	EM	$\mathbf{B}_2$	$\mathbf{B}_4$	$\mathbf{R}_2$	$\mathbf{R}_L$
L-Gen	47.3	81.2	73.6	80.9	86.3
L-Ptr-Gen	50.5	82.9	75.4	83.8	87.8
L-Ptr-Net	51.5	82.7	75.5	84.0	88.2
L-Ptr-λ	42.3	82.9	73.8	81.1	84.1
T-Gen	35.4	72.7	62.5	74.5	82.9
T-Ptr-Gen	53.1	84.4	77.6	85.0	89.1
T-Ptr-Net	53.0	83.9	77.1	85.1	88.7
T-Ptr-λ	52.6	85.6	78.1	85.0	89.0
RUN	66.4	**91.4**	86.2	90.4	93.5
MIUR (Ours)	**67.7**	91.2	**86.4**	**90.7**	**93.7**

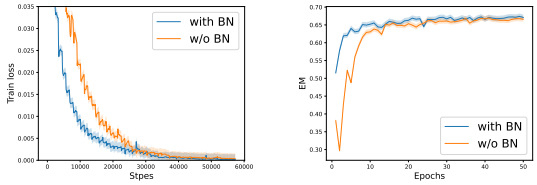
Table 7: Experimental results on REWRITE.

Model	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_4$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$
Copy	52.4	46.7	37.8	72.7	54.9	68.5
Rronoun Sub	60.4	55.3	47.4	73.1	63.7	73.9
L-Ptr-Gen	67.2	60.3	50.2	78.9	62.9	74.9
RUN	70.5	61.2	49.1	79.1	61.2	74.7
MIUR (Ours)	**71.3**	**63.4**	**51.7**	**81.6**	**64.5**	**77.4**

Table 8: Experimental results on CANARD.

D Effect of BatchNorm

To further explore the validity of BatchNorm for our model, we conducted controlled experiments on REWRITE. As shown in Figure 3, Figure 3(a) indicates the loss of training on REWRITE dataset with BN and without. Figure 3(b) shows the EM metrics of REWRITE validation set with BN and without. We can observe that the incorporation of BatchNorm after the construction of the joint feature matrix leads to faster convergence and enhances the model's ability to learn global semantic information efficiently.



(a) (b)

Figure 3: (a) shows the loss of training on REWRITE dataset with BatchNorm and without. (b) shows the EM metrics of REWRITE validation set with BatchNorm and without.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
5

☐ 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?
1

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

B ☑ Did you use or create scientific artifacts?

1

☒ B1. Did you cite the creators of artifacts you used?
While the paper itself does not include explicit citations to the creators of the artifacts used, the corresponding Git code repository's README.md file mentions the appropriate citations and attributions.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The code repository provided follows the Apache-2.0 license, which governs the terms and conditions for using and distributing the code.

☐ 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.
Left blank.

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

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ 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?
No response.

☐ 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.)?
No response.

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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.