MULTI-TURN INCOMPLETE UTTERANCE RESTORATION AS OBJECT DETECTION

*Wangjie Jiang**, *Siheng Li**, *Jiayi Li, Yujiu Yang*†

Tsinghua Shenzhen International Graduate School, Tsinghua University
{*jwj20, lisiheng21, lijy20*}*@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn*

ABSTRACT

In this paper, we investigate the task of multi-turn incomplete utterance restoration to tackle the issue of frequent coreference and information omission in multi-turn dialogues. Recent works mainly focus on edit-based approaches which have been proven to outperform traditional generation-based models in terms of accuracy and efficiency. However, they only model token-level edit relationships while ignoring span-level edit relationships. Our experiments find this breaks the semantic integrity of edit span, which causes inaccurate edit span prediction and disfluent utterance restoration. To address the problem, we propose a novel approach to directly model span-level edit relationships between the incomplete utterance and context. Specifically, we build an edit matrix in which each rectangular region represents a span-level edit operation. Then, we detect the region with a well-designed dual-branch detection module inspired by object detection. Empirical results demonstrate that our method outperforms state-of-the-art methods significantly on two public datasets. In addition, further studies verify that our method is capable of preserving the semantic integrity of edit span.

***Index Terms*—** Multi-turn dialogue, incomplete utterance restoration, text editing, object detection

1. INTRODUCTION

Dialogue systems have attracted increasing interest, and been applied to various scenarios, such as virtual assistants [1], customer service systems [2], etc. Despite the remarkable progress in dialogue systems, several challenges remain in the understanding of complex multi-turn dialogues. A major one is the frequently occurring coreference and information omission [3]. According to previous research [4], this phenomenon exists in more than 70% of the utterances in multi-turn dialogues. Taking the dialogue in Table 1 for example, the incomplete utterance $\mathbf{u}_3$ not only omits the object "玩电子游戏" (playing video games), but also refers to the original subject "李明" (Li Ming) via the pronoun "他" (he). Without expanding the coreference or omission to recover the full information, it is hard for machines to understand the real intention and generate coherent responses [5]. To tackle the problem, **M**ulti-turn **I**ncomplete **U**tterance **R**estoration (**MIUR**) [3, 4, 6] is proposed to restore the information of the incomplete utterance so that it can be understood without previous context. As shown in Table 1, by restoring the hidden semantics behind $\mathbf{u}_3$, $\mathbf{u}_3^r$ is more straightforward. Thus, MIUR can serve as a general pre-process to help multi-turn dialogue modeling [7].

Turn	Utterance (*Translation*)
$\mathbf{u}_1$ ($\mathbf{S}_1$)	很多学生喜欢玩电子游戏。 *Many students like playing video games.*
$\mathbf{u}_2$ ($\mathbf{S}_2$)	李明也喜欢么? *Dose Li Ming like it, too ?*
$\mathbf{u}_3$ ($\mathbf{S}_1$)	不，他不喜欢。 *No, he doesn't like.*
$\mathbf{u}_3^r$ ($\mathbf{S}_1$)	不，李明不喜欢玩电子游戏。 *No, Li Ming doesn't like playing video games.*

Table 1: An example of multi-turn dialogue between speaker $\mathbf{S}_1$ and $\mathbf{S}_2$, containing the context utterances $\mathbf{u}_1$ and $\mathbf{u}_2$, the incomplete utterance $\mathbf{u}_3$ and the restored utterance $\mathbf{u}_3^r$.

Traditional approaches [4, 6] usually formulate MIUR as an autoregressive text generation task and apply Seq2Seq architecture [8, 9] with copy mechanism. Specifically, Su et al. [4] propose a transformer-based generative model with pointer network to rewrite the incomplete utterance. Pan et al. [6] present a cascade frame of "pick-and-combine", in which the picking stage predicts the omitted words and the combining stage generates the restored utterance. Although these models achieve good performance, they suffer from low efficiency because of generating from scratch, and neglect the trait that the main structure of the restored utterance is always the same as the incomplete utterance.

In contrast, recent works [10, 11] focus on the text editing approach, as it serves as parallel edit operations on the incomplete utterance. Huang et al. [10] first predict the editing label of each token in the incomplete utterance and then edit with the corresponding edit operations. Liu et al. [11] utilize the word-level edit matrix to represent token-to-token edit operations and predict the edit matrix with semantic segmentation method. However, existing edit-based methods only model token-level edit relationships, while ignoring the semantic integrity of edit span. When dealing with the case in Table 1,

*Equal contribution.
†Corresponding author.

they would predict edit types of tokens "playing", "video" and "games" separately, while ignoring the span "playing video games" is an integrated semantic unit and should be treated as a whole. As a result, low accuracy on edit span prediction and severe disfluency on restored utterances have been observed in our empirical studies. [1]

To preserve the semantic integrity of edit span, we propose a novel method of modeling the multi-turn incomplete utterance **R**est**O**ration **A**s object **D**etection (**ROAD**), which directly models span-level edit relationships. Specifically, we build an edit matrix in which the rectangular regions represent span-level edit operations. Under this processing, we notice that operation identification is similar to the object detection task in computer vision. Therefore, we adopt a *Dual-branch Detection Module* to directly detect the edit operation which includes both edit type and edit span. In addition, we design an efficient strategy to obtain the final edit matrix and restore the incomplete utterance. Our contributions are as follows:

- To the best of our knowledge, this is the first work to directly model the span-level edit relationships between the incomplete utterance and context.
- We introduce a novel model ROAD that formulates MIUR as an object detection task and successfully predict the type and span of edit operation simultaneously.
- Experimental results on two benchmarks show that ROAD achieves state-of-the-art performance and is better at preserving the semantic integrity of edit span than baselines.

2. METHODOLOGY

2.1. Preliminary

Task Formulation. We denote a conversation session as $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_t\}$, in which $\mathbf{u}_i$ is the i-th utterance. Based on the dialogue context $\mathbf{c} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{t-1}\}$, our goal is to rewrite the incomplete utterance $\mathbf{x} = \mathbf{u}_t$ to a context-independent one $\mathbf{x^r} = \mathbf{u}_t^r$ which recovers all coreferenced and omitted information.

Edit Matrix. The edit matrix is denoted as $Y \in \mathbb{R}^{M \times N}$ which represents edit operations between M-length word sequence $\mathbf{c} = [c_i]_{i=1}^{M}$ and N-length word sequence $\mathbf{x} = [x_j]_{j=1}^{N}$. Three edit operations [11] are defined: *Substitute* means replacing the edit span in $\mathbf{x}$ with the corresponding edit span in $\mathbf{c}$; *Insert* aims to insert the edit span before a certain token in $\mathbf{x}$, and *None* represents no operation. Taking the edit matrix in Figure 1 for example, we can edit $\mathbf{x}$ by replacing $[x_2, x_3, x_4, x_5]$ with $[c_2, c_3, c_4]$ and insert $[c_7, c_8, c_9]$ before $[x_8]$.

Modeling as Object Detection. Object detection is a well-known computer vision task, its goal is to detect the type and bounding box of a specific object in an image, as in Figure 1.

[1] We reproduce the results of the best edit-based method [11], and find around 35% errors are caused by inaccurate edit span prediction, which further results in the disfluency of restored utterances.

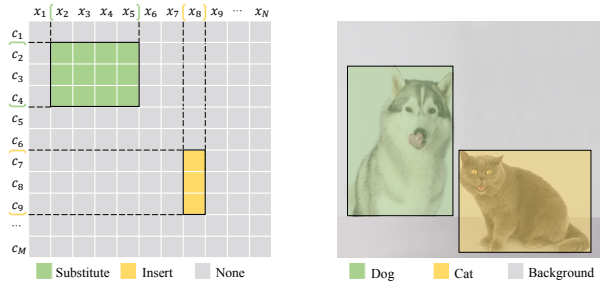

Fig. 1: Edit matrix (left) and object detection (right).

Similarly, our task is to recognize the edit type and edit span of a specific edit operation which is represented as the rectangular region in the edit matrix. This inspires us to identify the edit operation in a way analogous to object detection.

Overview. The framework of ROAD is shown in Figure 2. Given the input sequences, ROAD first employs a *Feature Map Construction Module* to construct a feature map which contains rich interaction features between the incomplete utterance and context. Based on this feature map, ROAD further utilizes a *Dual-branch Detection Module* which consists of a classification branch and a regression branch to generate the edit matrix. Concretely, the classification branch is used to detect the edit type, while the regression branch is used to detect the edit span. The outputs of the two branches are combined using a specific strategy to obtain the ultimate edit matrix. Finally, we edit the incomplete utterance with the obtained edit matrix and get the restored utterance.

2.2. Model Architecture

Feature Map Construction Module. This module is designed to encode the context $\mathbf{c}$ and the incomplete utterance $\mathbf{x}$ to obtain the feature map $\mathbf{F}$ that represents the token-to-token relevance between them. Given $\mathbf{c} = [c_i]_{i=1}^{M}$ and $\mathbf{x} = [x_j]_{j=1}^{N}$, we firstly concatenate them and leverage word embedding like GloVe [12] or contextualized word representation of BERT [13] to obtain the initialized representation for each token. Then we utilize the BiLSTM [14] to further capture the intra and inter relationships:

$$\begin{aligned} H &= \text{BiLSTM}(\text{Embedding}([c_1, ..., c_M, x_1, ..., x_N])) \\ &= [\tilde{c}_1, ..., \tilde{c}_M, \tilde{x}_1, ..., \tilde{x}_N], \end{aligned} \tag{1}$$

where $\tilde{c}_i$ and $\tilde{x}_i$ denote the context-aware representation of c_i and x_i respectively. To acquire the relevance features between them from multiple perspectives, we employ three similarity functions including element-wise similarity, cosine similarity and bi-linear similarity. Specifically, the relevance between c_i and x_j is calculated as follows:

$$\mathbf{F}_{i,j} = Sim(c_i, x_j) = \left[\tilde{c}_i \odot \tilde{x}_j; cos(\tilde{c}_i, \tilde{x}_j); \tilde{c}_i W \tilde{x}_j\right], \tag{2}$$

where W is a learnable weight matrix. Finally, we can obtain the feature map matrix $\mathbf{F} \in \mathbb{R}^{M \times N \times 3}$ which contains rich interaction information and is parallel to a multi-channel image.

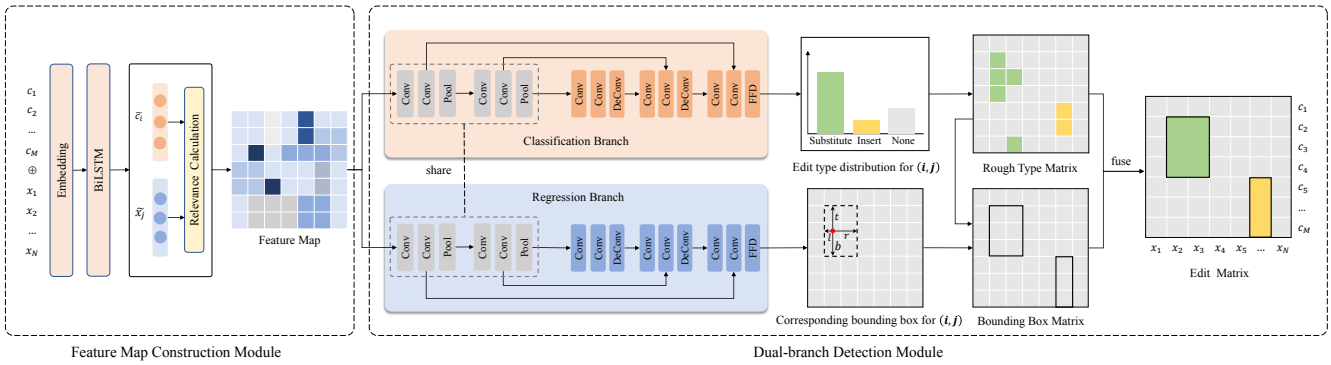

Fig. 2: The framework of our proposed model **ROAD**.

Dual-branch Detection Module. Taking the feature map $\mathbf{F}$ as the input, the dual-branch detection module is to produce the edit matrix $\mathbf{Y} \in \mathbb{R}^{M \times N}$, in which rectangular regions represent edit operations. As in Figure 2, this module consists of a classification branch and a regression branch. Each branch is structured as an encoder-decoder architecture, and the encoder is shared between branches to benefit from both subtasks. For each branch, we utilize a simplified U-Net [11, 15] as the backbone to capture the global interactions of the feature map, which contains two down-sampling blocks and two up-sampling blocks with skip connections. To be specific, each down-sampling block contains two separate convolution modules and a subsequent max pooling, and each up-sampling block consists of two separate convolution modules and a subsequent deconvolution layer. Furthermore, the number of channels is doubled in each down-sampling block, while halved in each up-sampling block. Finally a feed-forward neural network is employed to map the feature vector to the specific output. For each location (i, j), the output of the classification branch is $\boldsymbol{p}_{i,j} \in \mathbb{R}^3$ which is the probability distribution over three edit types, while the output of the regression branch is $\boldsymbol{t}_{i,j} = (l, t, r, b) \in \mathbb{R}^4$ which represents the distances from the location to the four sides of the corresponding predicted bounding box, as shown in Figure 2.

2.3. Training and Inference

Training. The two branches are jointly trained and the loss function is as follows:

$$\mathcal{L}(\{\boldsymbol{p}_{i,j}\}, \{\boldsymbol{t}_{i,j}\}) = \frac{1}{MN} \sum_{i,j} \mathcal{L}_{cls}(\boldsymbol{p}_{i,j}, \boldsymbol{c}^*_{i,j}) + \frac{\lambda}{MN} \sum_{i,j} \mathcal{L}_{reg}(\boldsymbol{t}_{i,j}, \boldsymbol{t}^*_{i,j}), \quad (3)$$

where $\mathcal{L}_{cls}$ is focal loss [16], $\mathcal{L}_{reg}$ is mean squared error and λ is the balance weight for the two parts. We construct the ground-truth edit matrix by distant supervision following [11]. Based on this, we further build the edit type label $\boldsymbol{c}^*_{i,j}$ and the distance label $\boldsymbol{t}^*_{i,j} = (l^*, t^*, r^*, b^*)$.

Inference. The inference is straightforward. Given $(\mathbf{c}, \mathbf{x})$, we forward it through the network and obtain the edit type distribution $\boldsymbol{p}_{i,j}$ and the distance vector $\boldsymbol{t}_{i,j}$ for each location (i, j). To get the target edit matrix, as in Figure 2, we first roughly decide the edit type of each location based on $\boldsymbol{p}_{i,j}$ and get Rough Type Matrix. In this matrix, we employ Hoshen-Kopelman algorithm [17] to find the connected regions, such as the four green cells in the upper left. Then, for all cells inside a connected region, we average their corresponding bounding boxes that can be inferred from $\boldsymbol{t}_{i,j}$ to obtain Bounding Box Matrix. Finally, combining Rough Type Matrix and Bounding Box Matrix, we can obtain the target Edit Matrix and thus restore the incomplete utterance.

3. EXPERIMENTS

3.1. Experiment Setting

Datasets. We carry out extensive experiments on two public datasets: (1) REWRITE [4]. A Chinese dataset collected from open-domain conversations. (2) CANARD [3]. An English dataset derived from question answering in context. These two are frequently used in previous studies.

Evaluation Metrics. We choose the same automatic metrics as in previous works [4, 10, 11], which contain BLEU [18], ROUGE [19] and Exact Match [4]. BLEU and ROUGE are widely used in generation tasks to measure the lexical similarity between generated utterance and ground-truth. Exact Match (EM) is a very strict metric which requires the restored utterance to be the same as the ground-truth.

Baselines. We compare the performance of our ROAD with the following methods: (1) **(L/T)-Ptr-λ** [4]. A Pointer-Generator Network based on LSTM/Transformer. (2) **SARG** [10]. A semi-autoregressive generator based on RoBERTa [20], which combines the sequence labelling for text editing and autoregression for text generation. (3) **RUN** [11]. RUN is a text editing approach based on a token-level edit matrix and achieves state-of-the-art performance on several benchmarks. For (L/T)-Ptr and RUN, we directly use the results in their original papers. For SARG, we reproduce the results with the

Model	EM	B_2	B_4	R_2	R_L
REWRITE					
L-Ptr-λ [4]	42.3	82.9	73.8	81.1	84.1
T-Ptr-λ [4]	52.6	85.6	78.1	85.0	89.0
RUN [11]	53.8	86.1	79.4	85.1	89.5
ROAD (Ours)	**54.8**	**87.2**	**80.4**	**85.5**	**90.0**
T-Ptr-λ + BERT [4]	57.5	86.5	79.9	86.9	90.5
SARG (RoBERTa based) [10]	64.8	89.4	84.3	88.8	93.4
RUN + BERT [11]	66.4	91.4	86.2	90.4	93.5
ROAD + BERT (Ours)	**69.0**	**91.8**	**87.0**	**91.1**	**94.0**
CANARD					
L-Ptr-λ [4]	16.5	60.3	50.2	62.9	74.9
RUN [11]	18.2	61.2	49.1	61.2	74.7
ROAD (Ours)	**18.7**	**62.8**	**50.6**	**63.5**	**76.8**
w/o Edit	12.2	46.7	37.8	54.9	68.5
ROAD w/o Cls.	17.6	61.6	47.8	61.8	74.2
ROAD w/o Reg.	18.1	61.4	49.0	62.3	74.7

Table 2: The experimental results on REWRITE and CANARD. $\mathbf{B}_X$, $\mathbf{R}_Y$ stand for BLEU-X, and ROUGE-Y respectively. "+BERT" means employing the BERT-enhanced embedding. "w/o Edit" means directly using the incomplete utterance as the restored utterance. "w/o Cls. (Reg.)" means that the classification (regression) branch is replaced by a feed-forward neural network with comparable parameters.

officially released code.

Implementation Details. The implementation of our model is based on PyTorch [21]. Specifically, the embedding dimension and hidden dimension in BiLSTM are 100 and 200 respectively. Adam [22] is used for optimizing and the learning rate is set as 1e-3, except for BERT as 1e-5. The batchsize is 16 when utilizing BERT while 24 otherwise. The balance weight λ is set as 1.0 in this paper. To be fair, we use $\text{BERT}_{\text{base}}$ as the context embedding layer when comparing with other BERT based methods.

3.2. Results

Table 2 shows the main results of ROAD and baselines. As for REWRITE, when comparing with LSTM/Transformer-based models, we employ randomly initialized embedding, while BERT-enhanced embedding is applied when comparing with methods using pretrained language models. ROAD outperforms all baselines significantly and reaches a new state-of-the-art performance. For the most challenging metric EM, ROAD gets a relative increase of $\mathbf{1.9}\%$ with randomly initialized embedding and $\mathbf{3.9}\%$ with BERT-enhanced embedding. It is worth noticing that ROAD gets more improvement when utilizing BERT, we conjecture this is because the contextualized word representation makes a greater difference when modeling the semantic integrity of edit span. Besides, ROAD also gains $\mathbf{2.7}\%$ relative improvement in EM on CANARD. In addition, the BLEU and ROUGE scores increase appreciably and the rate is $\mathbf{1.6}\%$ relatively on the two datasets. To sum up, these results demonstrate the superiority of our proposed model ROAD.

	IoU	PPL
RUN	0.389	4.705
ROAD	0.442	4.541
Ground-Truth	N/A	4.267

Table 3: Results of IoU and PPL comparison.

3.3. Analysis

Semantic Integrity. Except for the significant improvement in rewriting metrics, we conduct thorough experiments to prove that ROAD does better in preserving the semantic integrity of edit span compared with the best edit-based method RUN [11]. We evaluate the capacity of preserving edit span semantic integrity from two aspects: *edit span accuracy* and *fluency*. Specifically, edit span accuracy is localized, which directly measures the overlap between the predicted bounding box in edit matrix and ground-truth. Therefore, we employ IoU [23], a widely used metric in object detection, to measure the edit span accuracy. In addition, the fluency of the restored utterance measures semantic integrity globally. We choose Perplexity (PPL) [24] which is frequently used in text generation for fluency evaluation and employ GPT-2 [25] for calculation. As shown in Table 3, our method gets a remarkable improvement in both metrics [2], especially $\mathbf{13.6}\%$ improvement in IoU (edit span accuracy). We conjecture this is because ROAD is able to directly model span-level edit relationships, therefore leading to better semantic integrity.

Ablation Study. To prove the effectiveness of each component, we carry out ablation study as in Table 2. First, we can see that "Edit" is necessary according to the huge drop in "w/o Edit". Then, to show the effect of the Dual-branch Detection Module, we replace each branch with a feed-forward neural network with comparable parameters. The performance drops consistently in all metrics as in Table 2. This proves the necessity and rationality of each branch and also inspires more object detection methods for this task.

4. CONCLUSIONS

In this paper, we take the first step in modeling span-level edit relationships in MIUR. To be specific, we formulate this task as object detection and propose a novel model ROAD. Experimental results show that ROAD reaches a new state-of-the-art performance. In addition, the analysis also demonstrates the ability of our model to tackle the issue of edit span semantic integrity.

5. ACKNOWLEDGMENTS

This research was supported in part by National Key Research and Development Program of China (No. 2020YFB1708200) and the Shenzhen Key Laboratory of Marine IntelliSense and Computation under Contract ZDSYS20200811142605016.

[2] We conduct this experiment on the error cases of both RUN and ROAD.

6. REFERENCES

[1] Heung-Yeung Shum, Xiao-dong He, and Di Li, "From eliza to xiaoice: challenges and opportunities with social chatbots," *Frontiers of Information Technology & Electronic Engineering*, 2018.

[2] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al., "Alime assist: An intelligent assistant for creating an innovative e-commerce experience," in *CIKM*, 2017.

[3] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber, "Can you unpack that? learning to rewrite questions-in-context," in *EMNLP-IJCNLP*, 2019.

[4] Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou, "Improving multi-turn dialogue modelling with utterance rewriter," in *ACL*, 2019.

[5] Zhi Chen, Lu Chen, Hanqi Li, Ruisheng Cao, Da Ma, Mengyue Wu, and Kai Yu, "Decoupled dialogue modeling and semantic parsing for multi-turn text-to-SQL," in *Findings of ACL-IJCNLP*, 2021.

[6] Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu, "Improving open-domain dialogue systems via multi-turn incomplete utterance restoration," in *EMNLP-IJCNLP*, 2019.

[7] Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins, "Decontextualization: Making sentences stand-alone," *Transactions of the Association for Computational Linguistics*, 2021.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[9] Han Shu, Jiahao Wang, Hanting Chen, Lin Li, Yujiu Yang, and Yunhe Wang, "Adder attention for vision transformer," in *NeurIPS*, 2021.

[10] Mengzuo Huang, Feng Li, Wuhe Zou, and Weidong Zhang, "SARG: A novel semi autoregressive generator for multi-turn incomplete utterance restoration," in *AAAI*, 2021.

[11] Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang, "Incomplete utterance rewriting as semantic segmentation," in *EMNLP*, 2020.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning, "GloVe: Global vectors for word representation," in *EMNLP*, 2014.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[14] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, 1997.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[17] Joseph Hoshen and Raoul Kopelman, "Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm," *Physical Review B*, 1976.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[19] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *ACL*, 2004.

[20] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu, "Pre-training with whole word masking for chinese bert," *arXiv preprint arXiv:1906.08101*, 2019.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*. 2019.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[23] Md Atiqur Rahman and Yang Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *ISVC*, 2016.

[24] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *AAAI*, 2016.

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