Medical Dialogue Response Generation with Pivotal Information Recalling

Yu Zhao
zhao.yuhitsz@163.com
Harbin Institute of Technology
Shenzhen, China

Yunxin Li
liyunxin987@163.com
Harbin Institute of Technology
Shenzhen, China

Yuxiang Wu
yuxiang.wu@cs.ucl.ac.uk
University College London
London, United Kingdom

Baotian Hu†
hubaotian@hit.edu.cn
Harbin Institute of Technology
Shenzhen, China

Qingcai Chen
qingcai.chen@hit.edu.cn
Harbin Institute of Technology
Shenzhen, China

Xiaolong Wang
xlwangsz@hit.edu.cn
Harbin Institute of Technology
Shenzhen, China

Yuxin Ding
yxding@hit.edu.cn
Harbin Institute of Technology
Shenzhen, China

Min Zhang
zhangmin2021@hit.edu.cn
Harbin Institute of Technology
Shenzhen, China

ABSTRACT

Medical dialogue generation is an important yet challenging task. Most previous works rely on the attention mechanism and large-scale pretrained language models. However, these methods often fail to acquire pivotal information from the long dialogue history to yield an accurate and informative response, due to the fact that the medical entities usually scatters throughout multiple utterances along with the complex relationships between them. To mitigate this problem, we propose a medical response generation model with Pivotal Information Recalling (MedPIR), which is built on two components, i.e., knowledge-aware dialogue graph encoder and recall-enhanced generator. The knowledge-aware dialogue graph encoder constructs a dialogue graph by exploiting the knowledge relationships between entities in the utterances, and encodes it with a graph attention network. Then, the recall-enhanced generator strengthens the usage of these pivotal information by generating a summary of the dialogue before producing the actual response. Experimental results on two large-scale medical dialogue datasets show that MedPIR outperforms the strong baselines in BLEU scores and medical entities F1 measure.

CCS CONCEPTS

• Applied computing → Health care information systems; • Computing methodologies → Discourse, dialogue and pragmatics; Natural language generation.

†Both authors contributed equally to this research.
Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD’22, August 14–18, 2022, Washington, DC, USA
© 2022 Copyright held by the owner/authors(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9385-0/22/08...$15.00
https://doi.org/10.1145/3534678.3542674

1 INTRODUCTION

Medical dialogue system (MDS) has received much attention due to its high practical value. Previous works [5, 15, 21] usually model the dialogue history as sequential text and employ the sequence-to-sequence (Seq2Seq) models that built on large-scale pretrained text encoder and decoder to generate medical responses.

To have a comprehensive understanding of the patient, medical dialogues are always relatively long, and there are rich medical terminologies scattered in multiple utterances. Some works [9, 19, 20, 22] introduce the external medical knowledge into the Seq2Seq models and show that it can improve the performance. But these works fall short in utilizing the complex medical relationships between different utterances, which is important for inducing the next response. As shown in Figure 1, the entities tenesmus and enteritis indicate the symptom relationship between utterance#1 and utterance#4. Due to ignoring the medical relationship between utterances, the strong baseline model BERT-GPT-Entity [5] misses the pivotal entity colitis in the generated response. Our MedPIR derives the colitis from enteritis and generate a more accurate response.

How to acquire pivotal information from long dialogue history is the core of MDS. Previous works heavily rely on the cross-attention mechanism to use dialogue history, which falls short in locating the key information from a long sequence. This issue may be caused by the fact that the cross-attention mechanism is not trained with explicit supervision signals when recalling pivotal information. Recent works [8, 15, 26, 32] proposed to extract the medical key
最近总是拉红色糊状大便，有里急后重感。请问是什么原因？男，24岁。

#2 Doctor: 您好！这种情况多长时间了？（Hello, how long has this been going on?）

#3 Patient: 有一个多月了，隔几天就有这种情况。（It has been more than a month, and this happens every few days.）

#4 Doctor: 这种情况考虑是肠炎的可能。（It could be enteritis with such symptoms.）

#14 Patient: 这个检查完了，能知道是什么病因吗？现在特别担心是不是病情严重。（After examination, can we figure out what’s the cause? I’m really worried about my illness.）


doctor in the gastrointestinal department.)

是不是病很严重
好做一下结肠炎（It could be enteritis with such symptoms.）

现在特别担心这个检查完了，能知道是什么病因吗? (After examination, can we figure out what’s the cause? I’m really worried about my illness.)

Figure 1: An excerpted medical dialogue from MedDG [21]. The colored words are key medical phrases and the underlined parts represent the pivotal information to induce the response. The knowledge graph shown on the right is useful for diagnosing. The responses generated by the baselines and our proposed method are shown at the bottom.

2 RELATED WORKS

Medical Dialogue System (MDS). Previous MDS works mostly adopt a sequence-to-sequence framework [1, 30]. It consists of a context encoder to encode the dialogue history and a decoder to generate the response. Since the medical dialogue is often long and contains professional medical information, it is difficult for the attention mechanism to attend on the pivotal information in the dialogue history. To recognition key information in medical dialogues, Du et al. [8] and Zhang et al. [32] extract patient’s symptoms and medical status from history. Most recent, Li et al. [15] proposed a variational medical dialogue generation model strengthens by summarizing diagnosis history through a key phrase. However, these method only extract key information by phrases and cannot make fully use of the complicated pivotal information scattered in dialogue history. Different from previous works, we build medical dialogue graph that exploits medical relationship between utterances, and train the model to generate the pivotal information before producing the actual response, so that the model can learn to focus on the key information.

Dialogue Graph Construction. To model the relationship between utterances in a dialogue, Chen et al. [4], Sun et al. [28], Xu et al. [31] propose to construct a dialogue structure graph based on dialogue state transitions. Feng et al. [10] proposed to model the dialogue structure of the meeting by modeling different discourse relations. However, they did not exploit external knowledge base, which is essential for producing medical dialogue response. In contrast, we construct a knowledge-aware dialogue graph by incorporating external medical knowledge from CMeKG.

Knowledge-grounded Dialogue Generation. Recent works [6, 11, 17] proposed to improve the performance of dialogue modeling by retrieving relevant knowledge from the commonsense graph, such as ConceptNet [27], and incorporating the object facts in generation.
Dinan et al. [7], Kim et al. [13], Lian et al. [18], Zhao et al. [34] facilitated knowledge-ground dialogue generation by retrieving from unstructured documents. Li et al. [15] and Lin et al. [19] used medical knowledge graph to guide response generation through copy mechanism [24], but they did not use medical knowledge graph to exploit dialogue structure. In this work, the external knowledge is used to construct dialogue graph and is also encoded with a knowledge encoder.

3 METHODOLOGY
The key information of medical dialogue often scatters throughout the long history, making it difficult for traditional MDS models to acquire pivotal information from the dialogue history. In this section, we first describe the base medical response generation model in Section 3.1. Then, we introduce two techniques to improve the recalling of pivotal information from the dialogue – knowledge-aware dialogue graph encoder (Section 3.2) and recall-enhanced generator (Section 3.3). Finally, the training method of our proposed method is presented in Section 3.4.

3.1 Base Model
Most previous works in dialogue response generation [5, 21] adopt the sequence-to-sequence architecture to model the dialogue history and exploit external medical knowledge [15, 19, 20] to generate the response. For our base model, we follow Chen et al. [5] and use BERT-GPT as the backbone of our encoder and the generator. Given a dialogue history between a doctor and a patient \( X = (X_1, X_2, ..., X_M) \), where \( X_i = (x_{i1}, x_{i2}, ..., x_{i|X_i|}) \) is \( i \)-th utterance in the dialogue history with \( |X_i| \) tokens, the context encoder encodes the concatenation of utterances to obtain the context encoding \( H_{ctx} \).

We also follow previous works [7, 15, 32] to retrieve external knowledge and use a knowledge encoder to obtain the knowledge encoding \( H_k \) (more details are elaborated in Section 4.1.4). The base model produces responses \( Y = (y_1, y_2, ..., y|Y|) \) conditioned on both \( H_{ctx} \) and \( H_k \).

3.2 Knowledge-aware Dialogue Graph Encoder
Since the base dialogue model only views the medical dialogue history as a sequence of utterances, it is hard to model the diverse medical causal relationships between different utterances [10], which implies the pivotal medical information for inducing the next response. To tackle this problem, we propose the Knowledge-aware Dialogue Graph Encoder (KDGE) that constructs a dialogue graph with knowledge, and then encodes the graph with a graph attention network.

First, we transform the sequential dialogue history into a graph. Each utterance is regarded as a vertex, and there are two types of edge between the vertices. One type of edge connects the neighboring utterances, which denotes the normal temporal relations like previous works [4, 31]. The other type is knowledge-aware edge, which connects the scattered utterances with medical relationships. These knowledge-aware edges incorporates medical knowledge from external medical knowledge graph into the dialogues, allowing the model to represent complex medical relationships of the utterances. More concretely, we first extract medical entities from each utterance, and then look up the relationships between them from an external knowledge graph.\(^1\) We add a knowledge-aware edge between two utterances if there exists a relationship between the medical entities from the two utterances. Fig. 2 shows an example of this construction process. In the left part, the bold words are entities scattered in utterances, and the blue lines connect entities with certain relations. The right part represents the constructed knowledge-aware dialogue graph.

With the constructed knowledge-aware dialogue graph \( G \), we then apply Relational Graph Attention Network (RGAT) proposed by Busbridge et al. [2] to encode these pivotal relational information in the dialogue. For each vertex \( v_i \) in \( G \), we use a transformer-based encoder to encode its corresponding utterance, and compute the average of the token representations as the utterance embedding. Then the utterance embedding is concatenated with its speaker embedding (a trainable embedding that represents the role of the speaker) to form \( v_i \)'s initial vertex embedding \( v_i^0 \). At last, RGAT is used to compute the updated encoding of the vertices:

\[
(v_1, \ldots, v_M) = \text{RGAT} \left( (v_1^0, \ldots, v_M^0), G \right).
\]

(1)

To perform dialogue recalling, we regard the context encoding as initial history representation, and define recall score \( \alpha_{ti} \) as the importance of utterance \( X_i \) for recalling as follows:

\[
\alpha_{ti} = \sigma \left( (W^d_{ctx})^T (W^k v_i) \right),
\]

(2)

where \( h_{ctx} \) is mean-pooled from \( H_{ctx} \), \( W^d \) and \( W^k \) are trainable parameters, \( \sigma \) denotes the sigmoid function. Then the final structure encoding of \( X_i \) is obtained from the addition of utterance encoding \( h_i \) and vertex encoding \( v_i \) weighted by the corresponding recall score:

\[
h_{ctxi} = \alpha_{ti} (h_i + v_i).
\]

(3)

The concatenation of \( \{h_{ctxi}\}_{i=1}^M \) is the final structure encoding, denoted as \( H_{ctx} \).

3.3 Recall-Enhanced Generator
In the base model, the generator first performs unidirectional self-attention with the generated sequence to obtain the decoding state at each time, and then exploits \( H_{ctx} \) and \( H_k \) by the cross-attention mechanism. When this dialogue model is only trained to produce the response, its attention mechanism is often overwhelmed with

---

\(^1\)We choose CMeKG as our medical knowledge graph because it is the largest Chinese medical knowledge graph that is publically available.
the long dialogue history and fails to focus on the pivotal information. We propose Recall-Enhanced Generator (REG) to explicitly generate the pivotal information $R$ before producing the response. $R$ is a brief summary that contains key medical information of the dialogue history. After producing $R$, it will continue to generate focused response as follows:

$$y_t = REG(H_{ctx}, H_k, H_{stc}, [R; y_{<t}]).$$  

At training time, $R$ is automatically constructed with medical pretrained model PCL-MedBERT (more details introduced in Section 3.4) to serve as a supervision signal to train the model to recall pivotal information. At test time, MedPIR will first produce the recalled information and then generate the response. There are two main advantages of the method: 1) the qualified pre-generated recall $R$ provides a shortcut for the generator to access key history information through self-attention; 2) recalling strengthens the cross-attention mechanism to attend to the pivotal information provided by the encoders.

As shown in the right half of Fig. 3, tokens are first converted to embedding through the embedding matrix as the initial hidden state inputting to the generator. Then, REG sequentially generates the recalled pivotal information $R$, a separator, and finally the target response $Y$. Note that we use the average pooled knowledge encoding as the embedding of separator to drive the knowledge fusion during generation, as shown in the bottom-right part.

More specifically, REG consists of multiple layers decoder block. Let $h_{s,t-1}^{l-1}$ denote the output of $(l-1)$-th layer at $t$ step. The calculating process in $l$-th block can be formulated as:

$$h_{s,t}^l = \text{LayerNorm}\left(\text{SA}(h_{s,t-1}^l) + h_{s,t-1}^{l-1}\right),$$  

where $SA$ denotes unidirectional self-attention in decoder, and $\text{FFN}$ is a feed-forward network.

To integrate different type of information from the encoders, we introduce the $\text{Fusion}()$ operation, a gating mechanism that combines the context encoding $H_{ctx}$, structure encoding $H_{stc}$, and knowledge encoding $H_k$. It first condenses multifaceted encoding by taking $h_{s,t}^l$ as the query to perform cross-attention (CA) with $H_{ctx}$, $H_{stc}$ and $H_k$ respectively, and then conduct weighted sum of the condensed encodings with the gate scores:

$$\text{Fusion}(\cdot) = g_{ctx}^l \text{CA}^l(H_{ctx}, h_{s,t}^l) + g_k^l \text{CA}^l(H_k, h_{s,t}^l) + g_{stc}^l \text{CA}^l(H_{stc}, h_{s,t}^l),$$  

where the gate scores $g_{ctx}$, $g_{stc}$ and $g_k$ are obtained by a linear layer with sigmoid function:

$$g^l = \sigma(W^l \text{CA}^l(H, h_{s,t}^l)).$$  

Then, the three gate scores are normalized by the softmax function to obtain the final gate scores applied in Eq. (8). At the last layer, an output projection layer is applied to get the final generating distribution $p_t$ over vocabulary:

$$p_t = \text{softmax}\left(W_o h_{s,t}^l + b_o\right).$$  

While recalling pivotal information and generating response, the gate-based fusion network dynamically controls the inflows of context encoding, structure encoding, and knowledge encoding. The structure encoding obtained from KEDGE provides complementary information to the context encoding, facilitating REG to recall
We present the overall training algorithm in Algorithm (1).

### Algorithm 1: Training Algorithm

**Input:** training dialogue dataset \( \mathcal{D} \), initial parameter of MedPIR \( \theta \), initial learning rate \( \gamma \), PCL-MedBERT

1. **while not converged do**
2. 1. **foreach** sample \((X, y)\) in \( \mathcal{D} \) **do**
3. 3. Obtain \( X' \) by PCL-MedBERT;
4. 4. Calculate \( \{a_{\alpha_i}\}_{i=1}^{M} \) by Eq. (2);
5. 5. \( L_r \leftarrow \sum_{i=1}^{M} -r_i \log a_{\alpha_i} - (1 - r_i) \log (1 - a_{\alpha_i}) \);
6. 6. Calculate \( p(s_i | X, s_{<i}) \) and \( p(y_i | X, \mathcal{R}, y_{<i}) \) by Eq. (10);
7. 7. \( L_r \leftarrow \sum_{i=1}^{M} -\log p(s_i | X, s_{<i}) \);
8. 8. \( L_Y \leftarrow \sum_{i=1}^{M} -\log p(y_i | X, \mathcal{R}, y_{<i}) \);
9. 9. \( L \leftarrow \gamma L_Y + \lambda_r L_r + \lambda_L L_L \);
10. 10. \( \theta \leftarrow \theta - \gamma \nabla L \);
11. **end**
12. **end**

Then, we select \( k \) utterances with highest similarity scores, denoted as \( X' = (X_1', \ldots, X_k') \). The concatenation of \( X' \) is used as the target recall \( \mathcal{R} \) for training recall generation. Despite that this is a distantly-supervised method, the utterances extracted by PCL-MedBERT usually contain pivotal information for generating an informative medical response (see Fig. 5 for an example of extracted and generated recall sequence). To further facilitate the model to generate quality \( \mathcal{R} \) at inference, we also train it to identify pivotal utterances by supervising the recall score \( a_{\alpha_i} \) (obtained by Eq. (2)) through binary cross-entropy:

\[
L_r = \sum_{i=1}^{M} -r_i \log a_{\alpha_i} - (1 - r_i) \log (1 - a_{\alpha_i}),
\]

where \( r_i \in \{0, 1\} \) indicates whether \( X_i \) is in \( X' \). The higher \( a_{\alpha_i} \), the more important \( X_i \) for recalling.

### 3.4 Training

#### 3.4.1 Recall Supervision Signals

The ideal recall sequence \( \mathcal{R} \) is a summary of the current dialogue. But medical dialogue summary is not annotated in most cases. To deal with this problem, we introduce PCL-MedBERT to select the utterances that are most relevant to the target response as training signals. First, PCL-MedBERT encodes \( X_i \) and \( Y \) into \( h_i^X \) and \( h_i^Y \) respectively, and we use the cosine-similarity between them to score \( X_i \):

\[
sim(X_i, Y) = \frac{h_i^X \cdot h_i^Y}{\|h_i^X\| \|h_i^Y\|}
\]

The refined version of MedDialog preprocessed by Li et al. [15] to evaluate our method, where the training/development/test sets include 32723/3000/3000 dialogues respectively.

#### 4.1.3 Baselines

We use BLEU [23] to evaluate the n-gram lexical similarity, and use DISTINCT [16] to evaluate the diversity of the generated responses. We also take medical entities F1 score as an important metric, which can better evaluate the actuality of medical response than lexical similarity metrics. In MedDG dataset, we use the published script\(^3\) to recognize entities in responses, and evaluate different types of entity respectively. Due to MedDialog is not annotated with entities, we first collect medical entities in CMeKG, then extract entities in responses by string matching with the collected entities. Besides, we conduct human evaluation to evaluate the responses’ fluency, coherence, and correctness. The fluency only measures whether the generated response is fluent, while coherence measures whether the generated response is smooth and logical with context. The correctness evaluates whether the responses uses correct medical knowledge. Three metrics are scored by annotators with a range from 1 (bad) to 5 (excellent).

### 4 EXPERIMENTS

#### 4.1 Settings

**Datasets.** We adopt two medical dialogue datasets MedDG [21] and MedDialog [5] to evaluate our proposed model. Both of them are collected from online consultation websites. In MedDG, the training/development/test sets contain 14864/2000/1000 dialogues respectively, where each utterance is semi-automatically annotated with 5 types with a total of 160 medical entities. Li et al. [15] pointed that most dialogues in MedDialog have less than 5 utterances, which also contain few medical professional information. Thus, we follow

\[^{3}\] https://github.com/bwgkj/MedDG

[5] https://code.ihub.org.cn/projects/1775
| Model           | Sequence Metrics | Entity Metrics |
|-----------------|-----------------|---------------|
|                 | B@1  | B@2  | B@4  | D@2  | F1   | F1-D | F1-S | F1-A | F1-T | F1-M |
| Seq2Seq [29]    | 0.3852 | 0.3487 | 0.3297 | 0.8561 | 0.113 | 0.096 | 0.068 | 0.395 | 0.096 | 0.055 |
| Seq2Seq-Entity [21] | 0.3884 | 0.3416 | 0.3380 | 0.8635 | 0.195 | 0.224 | 0.159 | 0.406 | 0.178 | 0.107 |
| HRED [25]       | 0.3819 | 0.3365 | 0.3345 | 0.8670 | 0.109 | 0.097 | 0.064 | 0.383 | 0.098 | 0.053 |
| HRED-Entity [21] | 0.3942 | 0.3386 | 0.3255 | 0.8731 | 0.195 | 0.232 | 0.155 | 0.411 | 0.191 | 0.106 |
| DialoGPT [33]   | 0.3122 | 0.3125 | 0.3266 | 0.7869 | 0.122 | 0.100 | 0.089 | 0.409 | 0.104 | 0.094 |
| DialoGPT-Entity [21] | 0.3193 | 0.3106 | 0.3446 | 0.7892 | 0.176 | 0.180 | 0.095 | 0.366 | 0.203 | 0.094 |
| BERT-GPT [5]    | 0.4260 | 0.3593 | 0.3344 | 0.8893 | 0.146 | 0.138 | 0.099 | 0.399 | 0.106 | 0.101 |
| BERT-GPT-Entity [21] | 0.4286 | 0.3545 | 0.3187 | 0.8976 | 0.207 | 0.236 | 0.171 | 0.410 | 0.208 | 0.131 |
| VRBot [15]      | 0.3455 | 0.3144 | 0.3306 | 0.7460 | 0.075 | 0.073 | 0.052 | 0.194 | 0.100 | 0.035 |
| MedPIR (Ours)   | 0.4476 | 0.3866 | 0.3621 | 0.8915 | 0.227 | 0.263 | 0.175 | 0.413 | 0.213 | 0.144 |
| − Knowledge-aware dialogue graph encoder (KDGE) | 0.4109 | 0.3317 | 0.2888 | 0.8976 | 0.216 | 0.258 | 0.170 | 0.413 | 0.212 | 0.135 |
| − Recall-enhanced generator (REG) | 0.4247 | 0.3541 | 0.3353 | 0.8897 | 0.220 | 0.262 | 0.175 | 0.407 | 0.210 | 0.141 |
| − Knowledge encoder | 0.4379 | 0.3738 | 0.3573 | 0.8848 | 0.144 | 0.150 | 0.095 | 0.385 | 0.137 | 0.082 |
| − KDGE & REG    | 0.4023 | 0.3308 | 0.2964 | 0.8946 | 0.220 | 0.260 | 0.175 | 0.412 | 0.212 | 0.139 |

Table 1: Automatic evaluation results on MedDG dataset. The models with ‘-Entity’ suffix denotes their inputs incorporate entities by concatenating them with history directly. The entity F1 scores of different categories: F1-D (Disease), F1-S (Symptom), F1-A (Attribute), F1-T (Test) and F1-M (Medicine). B@n denotes BLEU-n and D@2 denotes DISTINCT-2.

4.1.4 External Knowledge. We exploit external knowledge following the previous knowledge-grounding dialogue generation methods [7, 15], where the retrieved knowledge is encoded and fused in the decoder. As verified by Liu et al. [21], predicting the medical entities used in the next response is helpful for informative response generation. Thus, we train our knowledge retrieval model to retrieve medical entities might be used in the response.

First, the medical entities appeared in the dialogue history are used as center nodes to select sub-graphs with one-hop relation in CMeKG. Then, we only retrieve entities contained in sub-graphs, which reduces the searching space for effective retrieval. Inspired by the bi-encoder dense retrieval method [12], we employ two independent PCL-MedBERT to encodes dialogue history \( X \) and any entity \( E \) (consists of several tokens) respectively, and take the representation at the [CLS] token as the encoder’s output.

Denote the dialogue history encoding as \( h_X \), and the entity encoding as \( h_E \), the inner product of \( h_X \) and \( h_E \) denotes the score to retrieve this entity. Let \( E^+_t \) be one of the positive entity appeared in the target response, alone with \( n \) negative entities \( \{ E^-_{t,j} \}_{j=1}^{n} \) not appeared. We optimize the loss function as the negative log likelihood of the positive entity:

\[
L_{X,E^+_t} = -\log \frac{\exp(h^+_X h^+_E)}{\exp(h^+_X h^+_E) + \sum_{j=1}^{n} \exp(h^+_X h^-_{E,j})}.
\]

The losses produced by all positive entities in each example are averaged as the final loss to train the retriever.

We retrieve top 20 entities for each dialogue history. This can be done with a single forward pass over datasets, where the retrieved entities are not changed during training and inference. Then, we employ another PCL-MedBERT as the knowledge encoder to encode the retrieved entities. The retrieved entities are sorted by their retrieval scores and are concatenated by [SEP] token to a sequence. The knowledge encoder encodes the sequence to knowledge encoding \( h_k \), and the encoder will be finetuned during training.

4.1.5 Implementation Details. For knowledge-aware graph encoder, the vertex embedding size and speaker embedding size is 512, and we use 2 layers RGAT [2] to encode the graph. For recall supervised signals construction, we set utterance number \( k \) to 3 in MedDG and 2 in MedDialog, and constrained the recall utterances in the last six rounds of dialogue history. For the RNN-based models, the encoder and decoder consist of one layer LSTM. Both the size of word embedding and hidden states are set to 300. For VRBot, we do not use the additional annotation of response intention for comparable experiments. For pre-trained models BERT-GPT and DialoGPT, the configurations are following the original works. We use exploitable pre-trained parameters of BERT-GPT to initialize our model. Due to its decoder uses encoding from encoder through self-attention, we initialize the cross-attention modules from scratch. We also pre-trained our model on medical domain corpus that used in BERT-GPT to improve the performance. For entity prediction in MedDG, we use 10-fold cross-validation models and ensemble results by majority voting method. The learning rate is initialized to \( 10^{-4} \) and \( 10^{-5} \) for the RNN-based model and pre-trained model. The loss coefficients \( \lambda_y \) and \( \lambda_k \) are set to 0.9, and \( \lambda_t \) is set to 0.1. We use the Adam optimizer [14], learning rate warm-up over the first 3000 steps and linear decay of the learning rate. Models generate responses through beam-sample algorithm, where beam-size and topk are set to 5 and 64. Other generation hyper-parameters keep default settings. We use the NLTK toolkit with SmoothFunction⁴ algorithm, where beam-size and topk are set to 5 and 64. Other generation hyper-parameters keep default settings. We use the NLTK toolkit with SmoothFunction⁴ algorithm, where beam-size and topk are set to 5 and 64. Other generation hyper-parameters keep default settings. We use the NLTK toolkit with SmoothFunction⁴ algorithm, where beam-size and topk are set to 5 and 64. Other generation hyper-parameters keep default settings. We use the NLTK toolkit with SmoothFunction⁴ algorithm, where beam-size and topk are set to 5 and 64. Other generation hyper-parameters keep default settings.
4.2 Results and Analysis

The automatic evaluation results are shown in Table 1 and Table 2. MedPIR outperforms other models both on BLEU and F1 metrics. As shown in Table 1, BERT-GPT-Entity is the model with the best all-around performance among comparative models. Our model outperforms the strongest baseline model BERT-GPT-Entity on BLEU-1/2/4 scores by a large margin, and outperforms it by 2 points on F1. In addition, MedPIR outperforms BERT-GPT$^*$ by 3 points on F1 and 1 points on BLEU-1 (see Table 2). These experimental results indicate that the proposed model is superior to previous models in terms of fluency and accuracy. We can see that transformer-based models DialoGPT, BERT-GPT$^*$ and MedPIR perform significantly better than RNN-based models, e.g. DialoGPT outperforms VRBot by 4 points on F1, suggesting the advantages of transformers-based models in larger dataset. Moreover, the experimental comparisons in DISTINCT-2 metric suggest our model reaches a competitive level in generating diverse responses when achieving new SOTA results on other evaluation metrics.

We also observe that all the models with -Entity improves the BLEU-1 and F1 scores. It verifies the medical entities are useful knowledge for medical response generation. But we also observe that the entity concatenation method is unstable, e.g., BERT-GPT-Entity obtain worse BLEU-4 than BERT-GPT. It may be caused by the low medical entities prediction accuracy. In addition, it is costly to annotate the entities entailed in utterances. But it is necessary for the entity concatenate method. By comparing the experimental results of MedPIR-KDGE & REG on F1 metric, we found that our knowledge retrieval strategy and gate-based fusion network are more effective and stable than other models.

| Model        | B@1 | B@2 | B@4 | D@2 | F1  |
|--------------|-----|-----|-----|-----|-----|
| Seq2Seq [29] | 0.301 | 0.225 | 0.163 | 0.791 | 0.063 |
| HRED [25]    | 0.299 | 0.226 | 0.180 | 0.785 | 0.080 |
| DialoGPT [33] | 0.275 | 0.204 | 0.155 | 0.706 | 0.128 |
| BERT-GPT$^*$ [5] | 0.298 | 0.232 | 0.202 | 0.821 | 0.145 |
| VRBot [15]   | 0.281 | 0.203 | 0.147 | 0.668 | 0.081 |
| MedPIR (Ours) | 0.308 | 0.237 | 0.210 | 0.811 | 0.174 |
| − KDGE       | 0.291 | 0.229 | 0.201 | 0.825 | 0.158 |
| − REG        | 0.285 | 0.229 | 0.202 | 0.813 | 0.163 |
| − Knowledge encoder | 0.296 | 0.231 | 0.202 | 0.817 | 0.164 |
| − KDGE & REG | 0.291 | 0.227 | 0.187 | 0.827 | 0.159 |

Table 2: Automatic evaluation results on MedDialog dataset. BERT-GPT$^*$ has been pre-trained on the MedDialog. REG indicates recall-enhanced generator, and KDGE indicates knowledge-enhanced dialogue graph encoder.

4.2.1 Ablation Study. We also take the ablation experiments to verify the effects of different modules in MedPIR, which are presented in the last four lines of Table 1 and Table 2. The experimental results suggest both knowledge-aware dialogue graph encoder (KDGE) and recall-enhanced generator (REG) improve the medical response generation. When we dropout the REG, where the generator produces responses directly, there is an obvious performance degradation on BLEU scores and a slight decrease on F1 score. It suggests the effectiveness of training the model generates pivotal information weakly supervised by PCL-MedBERT. When we only dropout the KDGE (− KDGE), the performance decrease significantly on BLEU and F1 scores. It indicates that the KDGE is vital to facilitate the recall-enhanced generator in MedPIR. Though the model is trained to generate recall, there is only a modest improvement without structure encoding. It is because the structure encoding captures the causal information from dialogue structure, supporting the model recalling long dialogue history effectively. Finally, when we dropout KDGE & REG, the performance decreases the most on BLUE metrics, indicating the effectiveness of the two main components in MedPIR.

As shown in Table 2, the REG and KDGE improve less in MedDialog than in MedDG. We suggest that it may be attributed to the fact that the length of dialogue in MedDialog is relatively short, which is also pointed by Li et al. [15]. The average number of utterances in MedDialog (9.5, the version cleaned by Li et al. [15]) is less than MedDG (21.5). It shows that MedPIR could focus on pivotal information scattered in long dialogue history and has preferable performance as the conversation length increases.

4.2.2 Analysis of Multifaceted Encoding. We select an example from MedDG and draw the picture to show how the model uses dialogue structure encoding, context encoding and knowledge encoding during recalling pivotal information and generating response. As shown in Fig. 4, the blue and red dots represent tokens of response and recall respectively. The horizontal axis and vertical axis show the gates’ scores $g_{stc}$ and $g_{ctx}$, respectively, and the scale of a dot is proportional to $g_k$. We observe that recall tokens distribute in the bottom-right part, and response tokens distribute in the upper-left part. It indicates that the model mainly uses structure encoding when recalling pivotal information and mainly uses context encoding when generating the response. This distribution shows that KDGE provides complementary information to the context encoding and facilitates REG to recall pivotal information. Though the response generation uses less structure encoding, the generator

![Figure 4: The blue dots and red dots represent tokens of response and recall respectively. The scale of the dot is proportional to the knowledge gate score.](image-url)
Dialogue History

X5: 这种情况考虑是肠炎的可能，最好是到医院消化内科就诊。
It could be enteritis with such symptoms. You had better to see a doctor in the gastrointestinal department.

X5: 8月份做过肠镜，说是有盲肠息肉，已经去掉了。之前的检查是因为拉肚子，痛、黏连。
Had done bowel mirror in August, there was caecal polyp, and I had taken it out. The previous test was done because of the bleeding. Now I’m pooping the red mushy stool all the time.

X6: 红色糊状物考虑病情已经加重，最好是到医院消化内科复查肠镜。
Red mushy stool means the illness is aggravated, you’d better to go to the hospital digestive department to review colonoscopy.

X8: 还要化验血常规，长期大便出血会引起贫血。
Still need to test blood routine. Long-term defective bleeds can cause anemia.

X10: 因为伴有里急后重，一定要注意查找病因。
Because accompanied by tenesmus, we must find the cause.

X14: 这个检查完了，能知道自己是什么病因吗？现在特别担心是不是病情较严重。
After the examination, can we figure out what’s the cause? I’m really worried about my illness.

Generated Recall

| 情况考虑是肠炎的可能，最好是到医院消化内科就诊。 | It could be enteritis with such symptoms. You had better to see a doctor in the gastrointestinal department. |
| 8月份做过肠镜，说是有盲肠息肉，已经去掉了。之前的检查是因为拉肚子，痛、黏连。 | Had done bowel mirror in August, there was caecal polyp, and I had taken it out. The previous test was done because of the bleeding. Now I’m pooping the red mushy stool all the time. |
| 红色糊状物考虑病情已经加重，最好是到医院消化内科复查肠镜。 | Red mushy stool means the illness is aggravated, you’d better to go to the hospital digestive department to review colonoscopy. |
| 还要化验血常规，长期大便出血会引起贫血。 | Still need to test blood routine. Long-term defective bleeds can cause anemia. |
| 因为伴有里急后重，一定要注意查找病因。 | Because accompanied by tenesmus, we must find the cause. |

Retrieved Knowledge

|  |  |  |
| --- | --- | --- |
| 检查 | 拉肚子 | 痛 |
| 症状 | 腹泻 | 黏连 |
| 概念 | 症状 | 进展 |

Generated Response

对话历史

X5: 这种情况考虑是肠炎的可能，最好是到医院消化内科就诊。

X5: 8月份做过肠镜，说是有盲肠息肉，已经去掉了。之前的检查是因为拉肚子，痛、黏连。

X6: 红色糊状物考虑病情已经加重，最好是到医院消化内科复查肠镜。

X8: 还要化验血常规，长期大便出血会引起贫血。

X10: 因为伴有里急后重，一定要注意查找病因。

X14: 这个检查完了，能知道自己是什么病因吗？现在特别担心是不是病情较严重。

生成的回应：

对话历史

X5: 这种情况考虑是肠炎的可能，最好是到医院消化内科就诊。

X5: 8月份做过肠镜，说是有盲肠息肉，已经去掉了。之前的检查是因为拉肚子，痛、黏连。

X6: 红色糊状物考虑病情已经加重，最好是到医院消化内科复查肠镜。

X8: 还要化验血常规，长期大便出血会引起贫血。

X10: 因为伴有里急后重，一定要注意查找病因。

X14: 这个检查完了，能知道自己是什么病因吗？现在特别担心是不是病情较严重。

生成的回应：

对话历史

X5: 这种情况考虑是肠炎的可能，最好是到医院消化内科就诊。

X5: 8月份做过肠镜，说是有盲肠息肉，已经去掉了。之前的检查是因为拉肚子，痛、黏连。

X6: 红色糊状物考虑病情已经加重，最好是到医院消化内科复查肠镜。

X8: 还要化验血常规，长期大便出血会引起贫血。

X10: 因为伴有里急后重，一定要注意查找病因。

X14: 这个检查完了，能知道自己是什么病因吗？现在特别担心是不是病情较严重。

生成的回应：

Table 3: Results of human evaluation. The maximum score of each indicator is 5.

| Model       | Fluency | Coherence | Correctness |
|-------------|---------|-----------|-------------|
| DialoGPT    | 3.69    | 3.46      | 2.76        |
| DialoGPT-Entity | 4.30    | 3.20      | 2.84        |
| BERT-GPT    | 4.36    | 3.73      | 3.06        |
| BERT-GPT-Entity | 4.35    | 3.74      | 3.13        |
| MedPIR      | 4.42    | 3.86      | 3.25        |

Figure 5: An example of recall and response generated by MedPIR in MedDG. The recalled utterances are colored correspondingly in the dialogue history. The bold entities in the history are used to retrieve knowledge. The retrieved knowledge with red-colored words are used in the generated response.
