

Semi-Supervised Variational Reasoning for Medical Dialogue Generation

Dongdong Li^{1*} Zhaochun Ren^{1†*} Pengjie Ren¹ Zhumin Chen¹
Miao Fan² Jun Ma¹ Maarten de Rijke^{3,4}

¹Shandong University, Qingdao, China

²Baidu Inc., Beijing, China

³University of Amsterdam, Amsterdam, The Netherlands

⁴Ahold Delhaize, Zaandam, The Netherlands

lddsdu@gmail.com, {zhaochun.ren, chenzhumin, majun}@sdu.edu.cn

jay.ren@outlook.com, fanmiao@baidu.com, M.deRijke@uva.nl

ABSTRACT

Medical dialogue generation aims to provide automatic and accurate responses to assist physicians to obtain diagnosis and treatment suggestions in an efficient manner. In medical dialogues two key characteristics are relevant for response generation: *patient states* (such as symptoms, medication) and *physician actions* (such as diagnosis, treatments). In medical scenarios large-scale human annotations are usually not available, due to the high costs and privacy requirements. Hence, current approaches to medical dialogue generation typically do not explicitly account for patient states and physician actions, and focus on implicit representation instead.

We propose an end-to-end variational reasoning approach to medical dialogue generation. To be able to deal with a limited amount of labeled data, we introduce both patient state and physician action as latent variables with categorical priors for explicit *patient state tracking* and *physician policy learning*, respectively. We propose a variational Bayesian generative approach to approximate posterior distributions over patient states and physician actions. We use an efficient stochastic gradient variational Bayes estimator to optimize the derived evidence lower bound, where a 2-stage collapsed inference method is proposed to reduce the bias during model training. A physician policy network composed of an action-classifier and two reasoning detectors is proposed for augmented reasoning ability. We conduct experiments on three datasets collected from medical platforms. Our experimental results show that the proposed method outperforms state-of-the-art baselines in terms of objective and subjective evaluation metrics. Our experiments also indicate that our proposed semi-supervised reasoning method achieves a comparable performance as state-of-the-art fully supervised learning baselines for physician policy learning.

*Equal contribution.

†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.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8037-9/21/07...\$15.00

<https://doi.org/10.1145/3404835.3462921>

CCS CONCEPTS

• **Applied computing** → Health care information systems; • **Computing methodologies** → Discourse, dialogue and pragmatics; • **Information systems** → *Specialized information retrieval*; *Users and interactive retrieval*.

KEYWORDS

Medical dialogue systems; Task-oriented dialogue generation; Variational inference; Semi-supervised learning

ACM Reference Format:

Dongdong Li, Zhaochun Ren, Pengjie Ren, Zhumin Chen, Miao Fan, Jun Ma, and Maarten de Rijke. 2021. Semi-Supervised Variational Reasoning for Medical Dialogue Generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3404835.3462921>

1 INTRODUCTION

Increasingly, conversational paradigms are being used to connect people to information, both to address open domain information needs [e.g., 14, 17, 23–25, 43, 50] and in support of professionals in highly specialized vertical domains [e.g., 48, 62]. Our focus is on conversational information seeking approaches in the medical domain. During clinical treatment, a conversational medical system can serve as a physician’s assistant to help generate responses for a patient’s need, i.e. inquire about symptoms, make a diagnosis, and prescribe medicine or treatment [54, 57, 59]. Intelligent medical dialogue systems (MDSs) are able to reduce the work pressure of physicians [46]. Previous work on MDSs mostly focuses on producing an accurate diagnosis given the dialogue context [32, 54, 57, 59]. There is very little work that considers the task of multi-turn medical dialogue generation to provide proper medical responses by tapping into large-scale medical knowledge sources.

There are two key characteristics that are specific to clinical decision support (CDS), and hence for dialogue systems that are meant to support clinical decision making: *patient states* (e.g., symptoms, medicine, etc.) and *physician actions* (e.g., treatments, diagnosis, etc.). These two characteristics make MDSs more complicated than other knowledge-intensive dialogue scenarios. Similar to task-oriented dialogue systems (TDSs), a medical dialogue generation (MDG) process can be decomposed into 3 stages: (1) *patient state tracking* (PST): after encoding the patient’s descriptions, the MDS

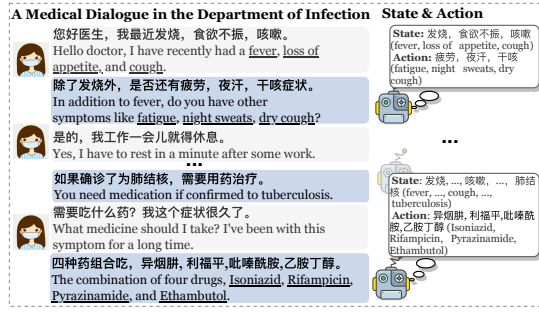


Figure 1: An example medical dialogue in the infection department, the left part shows the dialogue; the right part illustrates dialogue states and actions.

tracks the patient’s physiological condition, i.e., *patient states*, in the discourse context; (2) *physician policy learning* (PPL): given the patient’s states and utterances, the MDS generates the *physician’s action* to embed into the response; and (3) *medical response generation* (MRG): the MDS responds with a coherent sentence based on detected states and actions.

Figure 1 shows an example medical dialogue from the infection department. The left part lists the conversation, whereas the right part indicates patient states and physician actions during the conversation. At the first turn the patient shares their symptoms, i.e., *fever*, *loss of appetite*, and *cough*, as the patient state; the physician asks if the patient has other symptoms, i.e., *fatigue*, *night sweats*, and *dry cough*, to reflect the physician action at the second turn. Both states and actions vary as the conversation develops. At the last turn, the physician’s action is to prescribe drugs: *Isoniazid*, *Rifampicin*, *Pyrazinamide*, and *Ethambutol*.

The development of end-to-end MDG solutions faces a number of challenges: (1) Most TDSs need a large amount of manually labeled data to predict explicit dialogue states. In medical dialogues, annotators need medical expertise to annotate data. For privacy reasons, large-scale manually labeling intermediate states is problematic. Hence, few TDS methods can directly be applied to MRG [55]. (2) Existing approaches to MDG have a limited semantic understanding of the domain, which makes it hard to generate knowledgeable responses in a medical context [36]. (3) To help patients or physicians understand why a MDG system generates a response, explainability with indicative and interpretable information is indispensable, which is ignored by most TDS studies.

To address these challenges, we propose VRBot, which performs variational reasoning for MRG. Inspired by approaches to TDS, VRBot contains a *patient state tracker* and a *physician policy network* to detect patient states and physician actions, respectively. Unlike previous work, which learns from massive amounts of human-labeled observed variables, VRBot considers the patient state and the physician action as dual latent variables inferred in a variational Bayesian manner. We employ a stochastic gradient variational Bayes (SGVB) estimator to efficiently approximate the posterior inference. To alleviate the bias problem during SGVB estimation, we propose a 2-stage collapsed inference method to iteratively approximate the posterior distribution over states and actions.

To address the problem of limited semantic understanding during response generation, we proceed as follows. The physician policy network comprises an *action-classifier* that classifies physician

actions into *action categories*, and two reasoning components, a *context reasoning detector* and a *graph reasoning detector*, that infer explicit action keywords through the dialogue context and medical knowledge graph, respectively. With explicit sequences of patient states, physician actions, and multi-hop reasoning, VRBot is able to provide a high degree of explainability of its medical dialogue generation results.

To assess the effectiveness of VRBot, we collect a knowledge-aware medical dialogue dataset, KaMed. KaMed contains over 60,000 medical dialogue sessions with 5,682 entities (such as *Asthma* and *Atropine*). Using KaMed and two other MDG benchmark datasets, we find that VRBot, using limited amounts of labeled data, outperforms state-of-the-art baselines for MDG. Hence, given large-scale unlabeled medical corpora, VRBot can accurately trace the patient’s physiological conditions and provide more informative and engaging responses by predicting appropriate treatments and diagnosis. We also find that VRBot is able to provide more explainable response generation process over other MDG baselines.

Our contributions are as follows: (1) We propose an end-to-end medical response generation model, named VRBot. To the best of our knowledge, VRBot is the first to simultaneously model states and actions as latent variables in TDSs. (2) We devise a hybrid policy network that contains a context-reasoning detector and a graph-reasoning detector, which allow VRBot to predict physician actions based on the dialogue session and external knowledge simultaneously. (3) We show that VRBot can explicitly track patient states and physician actions even with few or no human-annotated labels. (4) We release KaMed, a large-scale medical dialogue dataset with external knowledge. (5) Experiments on benchmark datasets show that VRBot is able to generate more informative, accurate, and explainable responses than state-of-the-art baselines.

2 RELATED WORK

Medical dialogue systems. Previous methods for MDSs are modeled after TDSs, following the paradigm that a patient expresses their symptoms. Wei et al. [54] propose to learn a dialogue policy for automated diagnosis based on reinforcement learning. Lin et al. [32] build a symptom graph to model associations between symptoms to boost the performance of symptom diagnosis. Xu et al. [59] consider the co-occurrence probability of symptoms with diseases explicitly with reinforcement learning. Xia et al. [57] improve upon this work using mutual information rewards and generative adversarial networks. Meanwhile, various approaches have been explored to improve the understanding of medical dialogue histories, including symptom extraction [8], medical slot-filling [46], and medical information extraction [64]. Chen et al. [5] investigate the performance of pre-trained models for predicting response entities. Chen et al. [5] collect a dataset that consists of millions of dialogue sessions but do not explicitly consider learning the dialogue management as there are no human-annotated labels.

Currently, no prior work is able to explicitly learn a dialogue policy from a large-scale unlabeled corpus, greatly limiting the application of medical dialogue systems.

Dialogue state tracking. Dialogue state tracking plays an important role for TDSs. Conditional random field-based approaches [21, 22] and deep neural network-based approaches [12, 41] have been proposed to track states in modular TDSs [3]. Recently, end-to-end

TDSs have attracted a lot of interest [13, 17, 24, 30, 39, 56, 65]. For non-task-oriented dialogue generation, Serban et al. [44] and Chen et al. [4] propose generation methods with implicit state representations, for which it is hard to distinguish medical concepts. Dialogue states have also been represented as a sequence of keywords from the dialogue context [52]. Jin et al. [17] and Zhang et al. [65] propose semi-supervised generative models to leverage unlabeled data to improve state tracking performance. Liang et al. [29] propose an encoder-decoder training framework, MOSS, to incorporate supervision from various intermediate dialogue system modules. MOSS exploits incomplete supervision during model training. However, existing approaches fail to generate engaging and informative responses as do not address the semantic reasoning ability of the dialogue agents. As far as we know, no existing method simultaneously models states and actions under a few-shot regime.

In the MDG scenario, learning physician actions is as important as state tracking. Compared to [17, 29, 65], our model is capable of inferring missing states and actions simultaneously.

Knowledge-grounded conversations. The task of knowledge grounded conversation (KGC) is to generate responses based on accurate background knowledge. The task can be grounded into two categories according to the format of the background knowledge, i.e., structured KGC and unstructured KGC. The former focuses on exploiting knowledge triplets [35, 68] or knowledge graphs [15, 37, 49, 58, 67], the latter conditions on paragraph text [10, 18, 28, 40]. For structured KGC, Liu et al. [35] utilize a neural knowledge diffusion module to encode knowledge triplets to predict related entities. Liu et al. [37] augment a knowledge graph to integrate with dialogue contexts in an open-domain dialogue. Tuan et al. [49] assess a model’s ability to reason multiple hops using a Markov chain over a constructed transition matrix, so that the model can zero-shot adapt to updated, unseen knowledge graphs. Xu et al. [58] represent prior dialogue transition information as a knowledge graph and learn a graph grounded dialogue policy for generating coherent and controllable responses. Lei et al. [26] construct a user-item-attribute knowledge graph and ingeniously formalize dialogue policy learning as path reasoning on the graph.

Unlike most structured KGC methods that select knowledge from open-domain knowledge-bases, MDG aims to explore a multi-hop knowledge path transferred from patient states to physician actions using dedicated medical-domain knowledge graphs.

3 METHOD

3.1 Problem formulation

Medical dialogue systems. Given T dialogue turns, a medical dialogue session d consists of a sequence of utterances, i.e., $d = \{U_1, R_1, U_2, R_2, \dots, U_T, R_T\}$, where U_t and R_t refers to utterances from a patient and responses from a virtual physician, respectively. At the t -th turn, given the t -th patient utterance U_t and previous physician response R_{t-1} , the dialogue system generates a response R_t . Let $|U_t|$ be the number of words in U_t , we define $U_t = (U_{t,1}, U_{t,2}, \dots, U_{t,|U_t|})$ as a sequence of words. The full vocabulary is defined as \mathcal{V} . K denotes an external knowledge base in the medical dialogue system, where each triplet in K indicates a head entity, a relation, and a tail entity. Following [53], we construct a knowledge graph G^{global} by linking all triplets with overlapping

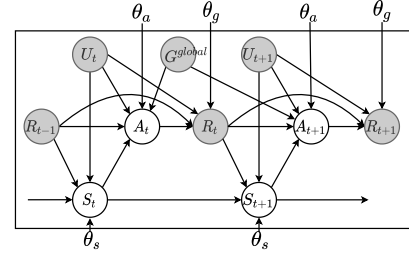


Figure 2: The graphical representation of VRBot. Shaded nodes represent observed variables.

entities (i.e., two triples will be linked iff they share overlapping entities) in K . We assume that each entity is categorized into a set of entity types, i.e., $E_{type} = \{\text{disease, symptoms, medicines, treatments}\}$.

We consider VRBot as a model with parameters θ . Given the dialogue context, responses, and the knowledge graph G^{global} , we aim to maximize the probability distribution over d in VRBot:

$$\prod_{t=1}^T p_{\theta}(R_t | R_{t-1}, U_t, G^{global}). \quad (1)$$

Patient states and physician actions. Text-span based dialogue state trackers have the double advantage of simplicity and good interpretability [17, 24, 55]. Hence, at the t -th turn, we define a text span S_t (i.e., a sequence of words) as the *patient state* to summarize past utterances and responses (i.e., $U_1, R_1, \dots, R_{t-1}, U_t$). Then we take S_t as constraints to search in a knowledge base. Similar to S_t , we also use a text span A_t to represent the *physician action* at the t -th turn, which summarizes the physician’s policy such as diagnose, medicine, or treatment. A_t is predicted through a policy learning process given S_t . Thus, task completion in MDG becomes a problem of generating two successive text spans, S_t and A_t , at each turn.

As text spans also help to improve the performance of response generation [17, 24], generating S_t and A_t at each turn is a key component in MDG. In this paper, the problem of MDG is decomposed into three successive steps: (1) generating a state span S_t ; (2) generating an action span A_t ; and (3) generating the response R_t .

Variational Bayesian generative model. Large volumes of intermediate annotations for patient states and physician actions are impractical in MDG. Thus, in VRBot we regard S_t and A_t as latent variables within a Bayesian generative model, so we reformulate Eq. 1 as:

$$\prod_{t=1}^T \sum_{S_t, A_t} p_{\theta_g}(R_t | R_{t-1}, U_t, S_t, A_t) \cdot p_{\theta_s}(S_t) \cdot p_{\theta_a}(A_t), \quad (2)$$

where $p_{\theta_g}(R_t | R_{t-1}, U_t, S_t, A_t)$ is derived using a *response generator*, and $p_{\theta_s}(S_t)$ and $p_{\theta_a}(A_t)$ are estimated through a *patient state tracker* and a *physician policy network*, respectively.

The graphical representation of VRBot is shown in Fig. 2, where shaded and unshaded nodes indicate observed and latent variables, respectively. We see that a dependency exists between two adjacent states. At t , S_t is derived depending on previous state S_{t-1} , response R_{t-1} , and utterance U_t ; subsequently, A_t is inferred using S_t , R_{t-1} , U_t , and G^{global} . Thus, we calculate $p_{\theta_s}(S_t)$ and $p_{\theta_a}(A_t)$ as:

$$\begin{aligned} p_{\theta_s}(S_t) &\triangleq p_{\theta_s}(S_t | S_{t-1}, R_{t-1}, U_t) \text{ (prior state tracker),} \\ p_{\theta_a}(A_t) &\triangleq p_{\theta_a}(A_t | S_t, R_{t-1}, U_t, G^{global}) \text{ (prior policy network),} \end{aligned} \quad (3)$$

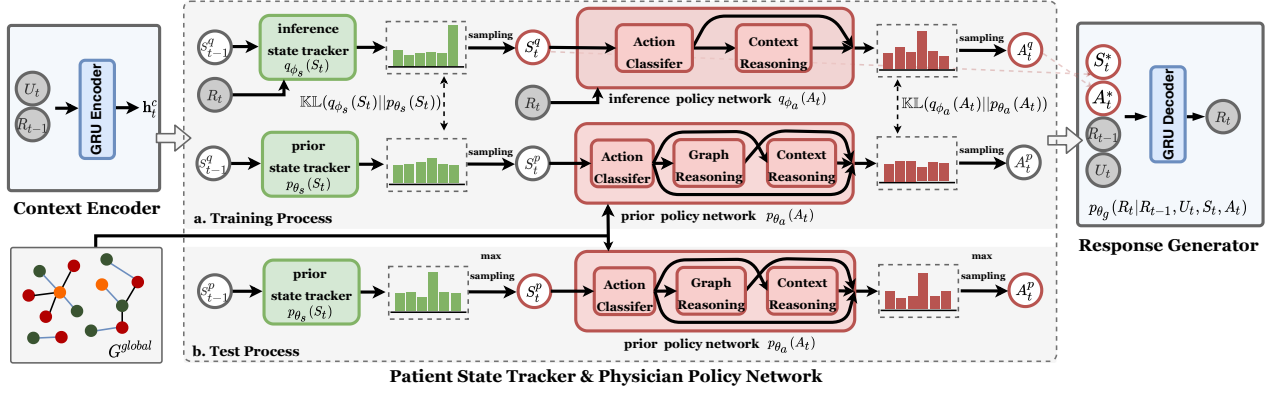


Figure 3: An overview of VRBot. We divide VRBot into a context encoder, a patient state tracker, a physician policy network, and a response generator. Labels a, b indicate different sampling procedure in training and test process respectively.

where θ_s and θ_a are parameters; and a fixed initial value is assigned to S_0 at the beginning. In VRBot we propose two *prior networks* to estimate probabilistic distributions in Eq. 3, i.e., a prior state tracker and prior policy network. Eventually, we draw a response R_t from $p_{\theta_g}(R_t|R_{t-1}, U_t, S_t, A_t)$, with parameters θ_g .

To maximize Eq. 2, we estimate the posterior distribution $p_{\theta}(S_t, A_t|R_t, R_{t-1}, U_t, G^{global})$. However, the exact posterior distribution is intractable due to its complicated posterior expectation estimation. To address this problem, we introduce two *inference networks* [20] (i.e., $q_{\phi_s}(S_t)$ and $q_{\phi_a}(A_t)$) to approximate the posterior distributions over S_t and A_t , respectively:

$$\begin{aligned} q_{\phi_s}(S_t) &\triangleq q_{\phi_s}(S_t|S_{t-1}, R_{t-1}, U_t, R_t) \text{ (inference state tracker),} \\ q_{\phi_a}(A_t) &\triangleq q_{\phi_a}(A_t|S_t, R_{t-1}, U_t, R_t) \text{ (inference policy network),} \end{aligned} \quad (4)$$

where ϕ_s and ϕ_a are parameters in inference networks.

Evidence lower bound (ELBO). At t , we derive the ELBO to optimize both prior and inference networks simultaneously as follows:

$$\begin{aligned} &\log p_{\theta}(R_t|R_{t-1}, U_t, G^{global}) \\ &\geq \mathbb{E}_{q_{\phi_s}(S_{t-1})} [\mathbb{E}_{q_{\phi_s}(S_t) \cdot q_{\phi_a}(A_t)} [\log p_{\theta_g}(R_t|R_{t-1}, U_t, S_t, A_t)] \\ &\quad - \mathbb{KL}(q_{\phi_s}(S_t) \| p_{\theta_s}(S_t)) - \mathbb{KL}(q_{\phi_a}(A_t) \| p_{\theta_a}(A_t))] \\ &= -\mathcal{L}_{joint}, \end{aligned} \quad (5)$$

where $\mathbb{E}(\cdot)$ is the expectation, and $\mathbb{KL}(\cdot \| \cdot)$ denotes the Kullback-Leibler divergence. To estimate Eq. 5, from $q_{\phi_s}(S_{t-1})$ we first draw a state S_{t-1}^q , which is for estimating $p_{\theta_s}(S_t)$ and $q_{\phi_s}(S_t)$; then, S_t^p is drawn from $p_{\theta_s}(S_t)$ and S_t^q is obtained through $q_{\phi_s}(S_t)$. We estimate $p_{\theta_a}(A_t)$ and $q_{\phi_a}(A_t)$ using S_t^p and S_t^q , respectively, and draw A_t^q from $q_{\phi_a}(A_t)$. Finally, $p_{\theta_g}(R_t|\cdot)$ generates R_t depending on S_t^q and A_t^q . The above sampling procedure is shown in Fig. 3 (a. Training process).

3.2 Context encoder

At t , we encode the dialogue history (R_{t-1}, U_t) into a list of word-level hidden vectors $H_t = (h_{t,1}, \dots, h_{t,|R_{t-1}|+|U_t|})$ using a bi-directional Gated Recurrent Unit (GRU) [6]:

$$H_t = \text{BiGRU}(h_{t-1}^c, e_1^{R_{t-1}}, e_2^{R_{t-1}}, \dots, e_{|R_{t-1}|}^{R_{t-1}}, \dots, e_{|U_t|}^{U_t}), \quad (6)$$

where $|R_{t-1}|$ and $|U_t|$ indicate the number of words in R_{t-1} and U_t respectively; $e_i^{R_{t-1}}$ denotes the embedding of the i -th word in R_{t-1} .

Initializing from the hidden representation h_{t-1}^c of the $(t-1)$ -th turn, the last hidden state $h_{t,|R_{t-1}|+|U_t|}$ attentively read H_t to get the t -th turn's hidden representation, i.e., h_t^c .

3.3 Patient state tracker

As we formulate patient states as text spans, the prior and inference state trackers are both based on an encoder-decoder framework. We encode S_{t-1}^q using a GRU encoder to get $h_{t-1}^{S_q}$ during the encoding procedure. We then incorporate $h_{t-1}^{S_q}$ with h_t^c to infer the prior state distribution $p_{\theta_s}(S_t)$ at the t -th turn. During the decoding procedure, we first infer the prior distribution over the patient state. We denote $b_{t,0}^{S_p} = W_s^p[h_t^c; h_{t-1}^{S_q}]$ as the initial hidden representation of the decoder, where W_s^p is a learnable parameter matrix, and $[\cdot; \cdot]$ denotes vector concatenation. At the i -th token during decoding, the decoder sequentially decodes S_t to output $b_{t,i}^{S_p}$ given previous token embedding $e_{t,i-1}^{S_p}$, next projects $b_{t,i}^{S_p}$ into the patient state space. We set S_t 's length to $|S|$, and the prior distribution over S_t is calculated as:

$$p_{\theta_s}(S_t) = \prod_{i=1}^{|S|} \text{softmax}(\text{MLP}(b_{t,i}^{S_p})), \quad (7)$$

where MLP is a multilayer perceptron (MLP) [9]. To approximate the state posterior distribution, the inference state tracker follows a similar process but additionally incorporates the encoding of R_t , i.e., h_t^R . The GRU decoder is initialized as $b_{t,0}^{S_q} = W_s^q[h_t^c; h_{t-1}^{S_q}; h_t^R]$, where W_s^q is a learnable parameter, and it outputs $b_{t,i}^{S_q}$ at the i -th decoding step. Accordingly, we write the approximate posterior distribution as:

$$q_{\phi_s}(S_t) = \prod_{i=1}^{|S|} \text{softmax}(\text{MLP}(b_{t,i}^{S_q})). \quad (8)$$

3.4 Physician policy network

The prior and inference policy networks are also based on an encoder-decoder structure. Specifically, we represent A_t as a pair of an action category A_t^c and a list of explicit keywords A_t^k , i.e., $A_t = \{A_t^c, A_t^k\}$. Here we set the length of A_t^k to $|A|$.

As for the prior policy network, at the beginning of the encoding procedure, we encode S_t^p to a vector \mathbf{h}_t^{SP} using a GRU encoder. Furthermore, external knowledge is important for the physician network to react given the patient state. As the external medical knowledge graph G^{global} is large (in the number of entities), we extract a sub-graph G_n^{local} from G^{global} via a knowledge base retrieval operation *qsub*, where we regard each entity in S_t^p as seed nodes during *qsub*. Starting from S_t^p , we extract all the accessible nodes and edges in G^{global} within n hops to get the sub-graph G_n^{local} [51]. Besides, we link all the entities appear in S_t^p to ensure G_n^{local} is connected.

To combine the relation type during information propagation, we employ the relational graph attention network (RGAT) [2] to represent each entity in the external knowledge graph. Given a graph $G = \{X, Y\}$ including relations Y and nodes X , after multiple rounds of propagation, RGAT outputs a feature matrix $G = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_X]$, where \mathbf{g}_x ($1 \leq x \leq X$) is the embedding of node x . We use RGAT to denote this operation, so we have: $G_n^{local} = \text{RGAT}(G_n^{local})$.

To decode outputs, we infer A_t^c and A_t^k sequentially. We devise an action classifier to infer A_t^c . Following [1], we compute an attention vector \mathbf{q}_t over G_n^{local} with \mathbf{h}_t^c as the query. Sequentially, the action classifier incorporates \mathbf{q}_t , and classifies physician action into four categories, i.e., *ask symptoms*, *diagnosis*, *prescribe medicine* and *chitchat*, as follows:

$$P_{\theta_{a,c}}(A_t^c) = \text{softmax}(\mathbf{W}_c^p[\mathbf{h}_t^{SP}; \mathbf{h}_t^c; \mathbf{q}_t]), \quad (9)$$

where \mathbf{W}_c^p is a learnable parameter. Then we draw an action category $A_t^{c,p}$ by sampling from $p_{\theta_{a,c}}(A_t^c)$.

A_t^k is decoded sequentially based on a GRU decoder. To infer the prior probabilistic distribution, two reasoning detectors (i.e., a context-reasoning detector and a graph-reasoning detector) are proposed to corporately project the hidden representation of the decoder to the action space at each decoding step. The decoder is initialized as $\mathbf{b}_{t,0}^{A^{k,p}} = \mathbf{W}_k^p[\mathbf{h}_t^{SP}; \mathbf{h}_t^c; \mathbf{e}_t^{A^{c,p}}]$, where $\mathbf{e}_t^{A^{c,p}}$ is the embedding of $A_t^{c,p}$. At the i -th decoding step, the decoder outputs $\mathbf{b}_{t,i}^{A^{k,p}}$. The context-reasoning detector and the graph-reasoning detector together infer $A_{t,i}^k$ with $\mathbf{b}_{t,i}^{A^{k,p}}$.

Learning from the raw context and state, the context-reasoning detector infers the prior distribution over $A_{t,i}^k$ using a MLP as follows:

$$p_{\theta_{a,d}}(A_{t,i}^k) = \frac{1}{z_A} \exp(\text{MLP}([\mathbf{h}_t^{SP}; \mathbf{h}_t^c; \mathbf{b}_{t,i}^{A^{k,p}}])), \quad (10)$$

where z_A is the normalization term shared with the graph-reasoning detector. The graph-reasoning detector considers to copy entities from G_n^{local} :

$$p_{\theta_{a,g}}(A_{t,i}^k) = \frac{1}{z_A} \mathbb{I}(e_j, A_{t,i}^k) \cdot \exp(\mathbf{W}_g[\mathbf{h}_t^c; \mathbf{b}_{t,i}^{A^{k,p}}; \mathbf{g}_j]), \quad (11)$$

where \mathbf{W}_g is a learnable parameter matrix, e_j is the j -th entity in G_n^{local} , \mathbf{g}_j is the j -th entry embedding of G_n^{local} , $\mathbb{I}(e_j, A_{t,i}^k)$ equals 1 if $e_j = A_{t,i}^k$ otherwise 0. Eventually, we calculate the prior distribution over A_t as follows:

$$p_{\theta_a}(A_t) = p_{\theta_{a,c}}(A_t^c) \cdot \prod_{i=1}^{|A|} [p_{\theta_{a,d}}(A_{t,i}^k) + p_{\theta_{a,g}}(A_{t,i}^k)]. \quad (12)$$

The inference policy network approximates the action category posterior distribution and keywords posterior distribution by extracting indicative information from the response R_t . A GRU encoder encodes R_t to \mathbf{h}_t^R , S_t^q to \mathbf{h}_t^{Sq} respectively. Then we get the action category approximate posterior distribution as follows:

$$q_{\phi_{a,c}}(A_t^c) = \text{softmax}(\mathbf{W}_c^q[\mathbf{h}_t^c; \mathbf{h}_t^{Sq}; \mathbf{h}_t^R]). \quad (13)$$

Thereafter, we draw $A_t^{c,q}$ via sampling from $q_{\phi_{a,c}}(A_t^c)$. To reinforce the effect of information from R_t , we only use the context-reasoning detector to approximate the posterior distribution of A_t^k . The decoder is initialized as $\mathbf{b}_{t,0}^{A^{k,q}} = \mathbf{W}_k^q[\mathbf{h}_t^c; \mathbf{h}_t^{Sq}; \mathbf{e}_t^{A^{c,q}}; \mathbf{h}_t^R]$, where $\mathbf{e}_t^{A^{c,q}}$ is the embedding of $A_t^{c,q}$, \mathbf{W}_k^q reflects a learnable parameter matrix.

At the i -th decoding step, the decoder outputs $\mathbf{b}_{t,i}^{A^{k,q}}$, so we have the approximate posterior distribution over the i -th action keyword:

$$q_{\phi_{a,d}}(A_{t,i}^k) = \text{softmax}(\text{MLP}([\mathbf{h}_t^c; \mathbf{h}_t^{Sq}; \mathbf{b}_{t,i}^{A^{k,q}}])). \quad (14)$$

Eventually, we get the approximate posterior distribution of A_t :

$$q_{\phi_a}(A_t) = q_{\phi_{a,c}}(A_t^c) \cdot \prod_{i=1}^{|A|} q_{\phi_{a,d}}(A_{t,i}^k). \quad (15)$$

Inspired by Jin et al. [17], we also employ the copy mechanism in $p_{\theta_s}(S_t)$ and $q_{\phi_s}(S_t)$, so as to copy tokens from R_{t-1} , U_t , S_{t-1}^q . In the same way, we copy tokens from R_t for $q_{\phi_a}(A_t)$.

3.5 Response generator

At the first stage during the response generation, we use a GRU encoder to encode S_t^q into S_t^q which is a word-level embedding matrix of S_t^q . Each column vector in S_t^q reflects an embedding vector of the corresponding word in S_t^q . In the same manner, $A_t^{k,q}$ is encoded to $A_t^{k,q}$. As mentioned in Sec. 3.3 and 3.4, we also calculate the holistic embedding \mathbf{h}_t^{Sq} and $\mathbf{h}_t^{A^{k,q}}$ from S_t^q and $A_t^{k,q}$, respectively. The response decoder with a GRU cell takes $\mathbf{b}_{t,0}^R = \mathbf{W}_d[\mathbf{h}_t^c; \mathbf{h}_t^{Sq}; \mathbf{e}_t^{A^{c,q}}; \mathbf{h}_t^{A^{k,q}}]$ as the initial hidden state.

At the i -th decoding step, the output $\mathbf{b}_{t,i-1}^R$ from the $i-1$ -th step attentively reads the context representation \mathbf{H}_t to get $\mathbf{b}_{t,i}^h$. Meanwhile, $\mathbf{b}_{t,i-1}^R$ attentively read S_t^q and $A_t^{k,q}$ to get $\mathbf{b}_{t,i}^s$ and $\mathbf{b}_{t,i}^a$ respectively. Subsequently, $[\mathbf{b}_{t,i}^h; \mathbf{b}_{t,i}^s; \mathbf{b}_{t,i}^a; \mathbf{e}_{t,i-1}^R]$ are fed into the decoder GRU cell to output $\mathbf{b}_{t,i}^R$, where $\mathbf{e}_{t,i-1}^R$ is the embedding of $(i-1)$ -st word in R_t . The probability of generating $R_{t,i}$ is formulated as a sum of the generative probability and a copy term:

$$\begin{aligned} p_{\theta_g}(R_{t,i}) &= p_{\theta_g}^g(R_{t,i}) + p_{\theta_g}^c(R_{t,i}), \\ p_{\theta_g}^g(R_{t,i}) &= \frac{1}{z_R} \exp(\text{MLP}(\mathbf{b}_{t,i}^R)), \\ p_{\theta_g}^c(R_{t,i}) &= \frac{1}{z_R} \sum_{j: W_j = R_{t,i}} \exp(\mathbf{h}_j^{W^\top} \cdot \mathbf{b}_{t,i}^R), \end{aligned} \quad (16)$$

where $p_{\theta_g}^g(R_{t,i})$ is the generative probability, $p_{\theta_g}^c(R_{t,i})$ is the copy term, z_R is the normalization term shared with $p_{\theta_g}^c(R_{t,i})$. We write W for a concatenation sequence of R_{t-1} , U_t , S_t^q , and $A_t^{k,q}$, where W_j is the j -th word in W , and \mathbf{h}_j^W is the j -th vector in $[\mathbf{H}_t; S_t^q; A_t^{k,q}]$.

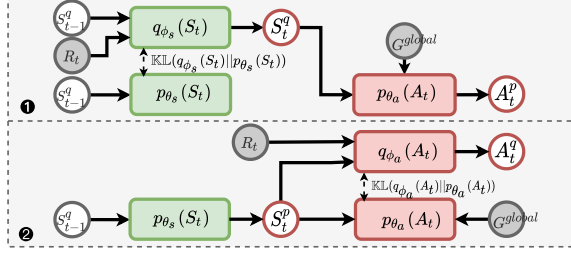


Figure 4: The graphical representation of 2-stage collapsed inference.

3.6 Collapsed inference and training

Eq. 5 provides a unified objective for optimizing all components. However, the joint distribution $p_{\theta_s}(S_t) \cdot p_{\theta_a}(A_t)$ is hard to optimize as $p_{\theta}(A_t)$ is easily misled with incorrect sampling results of S_t^p from $p_{\theta_s}(S_t)$. To address this problem, we propose a *2-stage collapsed inference* method by decomposing the objective function into 2-stage optimization objectives. During the first stage, we fit $p_{\theta_s}(S_t)$ to $q_{\phi_s}(S_t)$ to derive the ELBO (labeled by ❶ in Fig. 4):

$$\begin{aligned} & \log p_{\theta}(R_t | R_{t-1}, U_t, G^{global}) \\ & \geq \mathbb{E}_{q_{\phi_s}(S_{t-1})} \left[\mathbb{E}_{q_{\phi_s}(S_t)} \left[\mathbb{E}_{p_{\theta_a}(A_t)} [\log p_{\theta_g}(R_t | R_{t-1}, U_t, S_t, A_t)] \right] \right. \\ & \quad \left. - \mathbb{KL}(q_{\phi_s}(S_t) \| p_{\theta_s}(S_t)) \right] \\ & = -\mathcal{L}_s. \end{aligned} \quad (17)$$

Subsequently, similar to the optimization of ϕ_s and θ_s , $p_{\theta_a}(A_t)$ is fit $q_{\phi_a}(A_t)$ to formulate the ELBO (labeled by ❷ in Fig. 4) as follows:

$$\begin{aligned} & \log p_{\theta}(R_t | R_{t-1}, U_t, G^{global}) \\ & \geq \mathbb{E}_{q_{\phi_s}(S_{t-1})} \left[\mathbb{E}_{p_{\theta_s}(S_t)} \left[\mathbb{E}_{q_{\phi_a}(A_t)} [\log p_{\theta_g}(R_t | R_{t-1}, U_t, S_t, A_t)] \right] \right. \\ & \quad \left. - \mathbb{KL}(q_{\phi_a}(A_t) \| p_{\theta_a}(A_t)) \right] \\ & = -\mathcal{L}_a. \end{aligned} \quad (18)$$

Accordingly, the training procedure comprises two stages when no human-annotation exist. So we have:

$$\mathcal{L}^{un} = \begin{cases} \mathcal{L}_s & (1\text{st training stage}) \\ \mathcal{L}_s + \mathcal{L}_a & (2\text{nd training stage}). \end{cases} \quad (19)$$

We first minimize \mathcal{L}_s to get proper state tracking results. Then we jointly train all parameters to the 2nd stage optimization. We learn VRBot by SGVB and draw samples with the Gumbel-Softmax trick [16] to calculate the gradients with discrete variables.

If annotated states \tilde{S}_t and actions \tilde{A}_t are partially available, we add the auxiliary loss \mathcal{L}^{sup} to perform semi-supervised training:

$$\begin{aligned} \mathcal{L}^{sup} = & -(\log p_{\theta_g}(R_t | \tilde{S}_t, \tilde{A}_t, R_{t-1}, U_t) \\ & + \log(p_{\theta_s}(\tilde{S}_t) \cdot q_{\phi_s}(\tilde{S}_t)) + \log(p_{\theta_a}(\tilde{A}_t) \cdot q_{\phi_a}(\tilde{A}_t))). \end{aligned} \quad (20)$$

In the test process, we only execute $p_{\theta_s}(S_t)$ and $p_{\theta_a}(A_t)$ to infer patient states and physician actions (labeled by ❷ in Fig. 3).

4 EXPERIMENTAL SETUP

4.1 Research questions

We seek to answer the following research questions: (RQ1) How does VRBot perform on medical dialogue generation? Is unlabeled data helpful for generating accurate responses? (RQ2) What is the effect of each component in VRBot? Are the reasoning detectors helpful to improve physician action prediction? (RQ3) What is the

effect of the length of the patient state and physician action in VRBot? (RQ4) Can VRBot provide interpretable results?

4.2 Datasets

We adopt three medical dialogue datasets for our experiments, all of which are collected from real-world medical consultation websites after data anonymization, i.e., close to clinically authentic medical scenarios. Two have been applied in previous studies, and we propose a new dataset with large-scale external knowledge.

Existing medical dialogue datasets have a limited amount of external knowledge, a limited length of dialogues, and a handful of medical departments. These constraints make it difficult to evaluate MDG approaches. To address this problem, we collect a large-scale dataset **Knowledge-aware Medical conversation dataset (KaMed)** from ChunyuDoctor,¹ a large online Chinese medical consultation platform. The dataset caters for challenging and diverse scenarios, as it contains over 100 hospital departments with a large-scale external knowledge graph. To simulate realistic clinical conversational scenarios, in KaMed the average number of rounds of a dialogue is 11.62, much longer than existing medical dialogue datasets. Unlike other medical dialogue datasets, KaMed is equipped with large-scale external medical knowledge, crawled from CMeKG,² the largest Chinese medical knowledge platform.

To evaluate VRBot, we also use two benchmark datasets. MedDG [36] is collected from ChunyuDoctor, related to 12 types of common gastrointestinal diseases, and provides semi-automatic annotated states and actions; the average number of rounds of a dialogue session is 9.92. MedDialog [5] is collected from an online medical platform. We filter out dialogues with fewer than three rounds, but the average number of rounds is still relatively low, only 4.76. We also collect relevant medical knowledge for the MedDG and MedDialog datasets. The dataset statistics are listed in Table 1.

4.3 Baselines and comparisons

In the context of RQ1, we write VRBot\un for the model that is only trained using annotated data. We devise a variation of VRBot by replacing the GRU encoder with Bert, and use VRBot-Bert to denote it. In the context of RQ2, we write VRBot\S for the model that eliminates the latent state variable, VRBot\A for the model that eliminates the latent action variable, VRBot\G for the model without the graph-reasoning detector, VRBot\C for the model without the context-reasoning detector, and VRBot\2s for the model without 2-stage collapsed inference (i.e., minimizing \mathcal{L}_{joint} in Eq. 5).

As far as we know, only Liu et al. [36] have addressed the same task as we do. Thus, for MDG, we use HRED-Bert [36] as a baseline, which integrates Bert [7] with the HRED model for MRG. We consider three types of baseline: open-domain dialogue generation, knowledge grounded conversations, and task-oriented dialogue generation. As open-domain approaches, we use Seq2Seq [47], HRED [44], and VHRED [45] as baselines. Seq2Seq is a sequence-to-sequence generation model with attention and copy mechanism [11]; HRED uses a hierarchical encoder-decoder structure to model the dialogue at the word- and utterance-level; VHRED extends HRED with a continuous latent variable to facilitate generation. As knowledge-grounded methods, we use CCM [67], NKD [35], and PostKS [28] as baselines. CCM applies two graph attention

¹<https://www.chunyuyisheng.com/>

²<http://zstp.pcl.ac.cn:8002>

Table 1: Statistics of KaMed, MedDialog and MedDG; ✓ under ‘An’ indicates the dataset provides annotations.

Dataset	Train/Valid/Test	Entity/Triplet	Turn	An
KaMed	57,754/3,000/3,000	5,682/53,159	11.62	✗
MedDialog	32,723/3,000/3,000	4,480/79,869	4.76	✗
MedDG	14,864/2,000/1,000	160/1,240	9.92	✓

mechanisms to augment the semantic information during response generation; NKD uses a neural knowledge diffusion module to retrieve relevant knowledge; PostKS uses dialogue context and responses to infer the posterior knowledge distribution. For task-oriented dialogue generation, we use SEDST [17], LABES [65], MOSS [29], and DAMD [66] as baselines. SEDST formalizes the dialogue state as a text-span to copy keywords from question to state; LABES regards the state as discrete latent variables to conduct the Straight-Through Estimator for calculating the gradient; MOSS incorporates supervision from various intermediate dialogue system modules; DAMD uses GRU-based decoders to decode the state, action, and response in a supervised manner. Similar performance can be also observed in Transformer based methods [60, 61].

4.4 Evaluation metrics

Automatic evaluation. To assess the language quality for the generated responses, we employ classical word-overlap based metrics, *BLEU-2* (B@2) [42] and *ROUGE-2* (R@2) [31], to measure performance. As shortcomings have been reported for using BLEU/ROUGE to measure dialogue generation [33], we also use *Distinct-1* (D@1) and *Distinct-2* (D@2) [27], where Distinct-n is defined as the proportion of distinct n-grams in generated responses. To measure the correctness of prediction results from the physician policy network, following [36], we calculate *Precision* (P), *Recall* (R), and *F1* (F1) scores of predicted entities in the responses. We adopt the prefix *ma-* and *mi-* to indicate macro-average and micro-average Precision, Recall, and F1 scores, respectively. We also employ embedding-based topic similarity metrics [34], i.e., *Embedding Average* (EA) and *Embedding greedy* (EG), to evaluate the semantic relevance of the predicted entities between generated response and target response. We use mean explainability precision (MEP) and mean explainability recall (MER) to evaluate explainability [63].

Human evaluation. We randomly sample 600 dialogues and their corresponding generations from our model as well as the baselines. We recruit three professional annotators from a third-party hospital to evaluate the responses generated by different models. Following Liu et al. [36], we evaluate the responses generated by all models in terms of following three metrics: *sentence fluency* (Flu), *knowledge correctness* (KC), and *entire quality* (EQ). Flu measures if the generated response is smooth; KC evaluates whether the response is correct; EQ measures the annotator’s satisfaction with the generated response. Three annotators are asked to rate each generated response with a score range from 1 (bad) to 5 (excellent) for each entry. Model names were masked out during evaluation.

4.5 Implementation details

We conduct our experiments with a batch size of 16, and the size of embedding and the GRU hidden state set to 300 and 512, respectively. We set $n = 2$ in the *qsub* operation. The graph hidden size and output size are set to 128 and 512, respectively. All modules are trained in an end-to-end paradigm. We use the pkuseg [38] toolkit to

segment words. The vocabulary size is limited to 30,000 for KaMed and MedDialog, and 20,000 for MedDG. The lengths of the state text span and action text span are set to 10 and 3 in our experiments. We employ the PCL-MedBERT³ embedding which is trained on a large-scale medical corpus. We set the temperature of Gumbel-Softmax to $\tau = 3.0$, and anneal to 0.1 in 30,000 training steps. We use the Adam optimizer [19]; the learning rate is initialized to $1e^{-4}$ and decrease to $1e^{-5}$ gradually. For all models, we apply beam search decoding with a beam size of 5 for response generation.

5 EXPERIMENTAL RESULTS

5.1 Overall performance

We show the automatic evaluation results for all unsupervised models on KaMed and MedDialog in Table 2, and the semi-supervised results in Table 3. We see in Table 2 that VRBot significantly outperforms all baselines in terms of most evaluation metrics on both datasets. In terms of D@1 and D@2 VRBot outperforms other baselines as the generated responses in VRBot are more diverse. For KaMed, VRBot achieves an increase of 14.68%, 36.81%, 61.00%, and 67.57% over PostKS in terms of B@2, R@2, D@1, and D@2, respectively. For MedDialog, VRBot gives an increase of 21.47%, 14.17%, 31.29%, and 43.63% over PostKS. Models without reasoning give high *ma-P* and *mi-P* scores, but they do not perform well in terms of *ma-R*, *mi-R*, *ma-F1*, *mi-F1*. In terms of *ma-R*, *mi-R*, *ma-F1*, *mi-F1*, EA and EE, VRBot outperforms all baselines by a large margin. Hence, VRBot is effective in predicting physician actions. Table 3 shows the performance in semi-supervised settings on the MedDG dataset. SEDST and LABES are two state-of-the-art semi-supervised state tracking approaches. Without labeled action data, VRBot still achieves 23.35% and 26.73% improvements over LABES in terms of *ma-F1* and *mi-F1* with 25% labeled states; when the state labeling proportion increases to 50%, VRBot achieves an increase of 20.11% and 20.38%. VRBot outperforms VRBot\un by 12.36% and 10.36% in terms of *ma-F1* and *mi-F1* with the supervision proportion set to 50%; the increase is more significant with a lower supervision proportion. Thus, unlabeled data improves the performance of VRBot. VRBot outperforms MOSS by a large margin despite the fact that MOSS can also use unlabeled data; it outperforms MOSS by 11.90% and 14.14% in terms of *mi-F1* when with 25% and 50% labeled data, respectively. VRBot significantly outperforms HRED-Bert in terms of all metrics when the supervision proportion $\leq 25\%$. With 50% and 100% annotated data, VRBot-Bert still outperforms HRED-Bert by 12.04% and 7.93% in terms of *mi-F1*.

In Table 4, we perform a human evaluation on the KaMed and MedDG dataset to investigate the unsupervised and semi-supervised performance of VRBot. VRBot achieves the best performance in terms of all metrics on both datasets. On KaMed, VRBot outperforms SEDST and PostKS in terms of KC and EQ by a large margin. The result is consistent with our automatic evaluation results, confirming the importance of simultaneously modeling patient state and physician action. On MedDG, MOSS slightly outperforms DAMD, a fully-supervised method. VRBot achieves a 13% and 15% increase over MOSS in terms of KC and EQ. Thus, the unlabeled states and actions inferred by VRBot help to improve performance. We compute the average pairwise Cohen’s kappa (κ) to measure the consistency between annotators, and find that $0.6 \geq \kappa \geq 0.4$ for all metrics.

³https://code.iHub.org.cn/projects/1775/repository/mindspore_pretrain_bert

Table 2: Automatic evaluation on the KaMed and MedDialog datasets. Boldface scores indicate best results, significant improvements over the best baseline are marked with * (t-test, $p < 0.05$).

Dataset	Model	B@2	R@2	D@1	D@2	ma-P	ma-R	ma-F1	mi-P	mi-R	mi-F1	EA	EG
KaMed	Seq2Seq	2.71	1.58	1.24	6.85	24.82	11.14	15.38	27.60	12.78	17.47	27.93	27.75
	HRED	2.59	1.59	1.17	6.65	27.14	11.28	15.94	28.36	12.82	17.65	27.79	27.75
	VHRED	2.49	1.55	1.15	6.42	28.65	11.18	16.08	28.36	12.61	17.46	27.44	27.36
	CCM	2.42	1.47	0.51	2.12	19.05	10.59	13.61	23.53	14.49	17.93	33.64	33.44
	SEDST	2.40	1.39	0.43	2.19	26.45	10.92	15.46	28.86	13.81	18.68	31.85	31.64
	PostKS	2.52	1.44	1.00	5.55	25.18	11.34	15.64	26.01	14.15	18.33	29.76	29.61
	VRBot	2.89	1.97*	1.61	9.30	22.91	17.00*	19.52*	26.35	18.84*	21.97*	43.18*	43.11*
MedDialog	Seq2Seq	3.13	1.11	1.62	8.11	23.73	8.54	12.56	25.65	8.97	13.29	20.53	21.95
	HRED	2.56	0.85	1.72	8.54	23.38	8.77	12.75	25.34	8.79	13.06	20.64	21.97
	VHRED	2.82	1.01	1.74	8.84	26.23	8.87	13.26	26.29	9.00	13.41	20.15	21.43
	CCM	3.29	1.14	1.42	6.91	20.36	9.49	12.94	20.68	10.82	14.21	26.51	27.81
	SEDST	2.37	0.89	0.72	3.13	22.82	8.00	11.85	24.81	8.06	12.17	20.13	21.23
	PostKS	3.26	1.27	1.63	8.48	25.53	9.81	14.17	21.60	10.15	13.81	24.38	25.76
	VRBot	3.96*	1.45*	2.14	12.18	22.77	14.11*	17.42*	23.50	14.73*	18.11*	34.51*	36.79*

Table 3: Automatic evaluation on the MedDG dataset. S-Sup and A-Sup indicate the supervision proportion of states and actions, respectively. Models that are able to use unlabeled data are marked with #.

Model	S-sup	A-sup	ma-F1	mi-F1	EA	EG
SEDST#	25%	0%	13.50	20.94	22.39	33.19
	50%		12.29	19.97	23.12	33.59
LABES#	25%	0%	12.80	20.05	23.31	33.20
	50%		12.28	20.02	23.40	33.88
NKD	0%	25%	5.45	16.40	20.46	29.64
		50%	6.15	18.89	22.10	32.03
		100%	8.92	21.68	23.77	34.48
PostKS#	0%	25%	9.33	22.07	24.04	34.89
		50%	9.44	22.34	24.55	35.58
		100%	9.68	22.19	24.90	36.08
HRED-Bert	0%	10%	8.74	18.15	23.21	33.47
		25%	12.24	22.57	26.17	37.92
		50%	14.91	23.83	27.36	39.58
		100%	15.52	25.57	28.42	41.13
DAMD	25%	25%	12.94	21.62	23.91	34.82
	50%	50%	14.26	23.70	24.83	36.06
	100%	100%	13.47	25.06	26.28	38.39
MOSS#	25%	25%	12.74	23.03	25.83	37.46
	50%	50%	13.78	23.33	25.36	36.87
	100%	100%	13.84	24.36	25.21	36.69
VRBot\un	25%	25%	10.86	20.74	24.49	35.69
	50%	50%	13.10	24.13	26.11	38.19
VRBot#	25%	0%	15.79*	25.41*	24.09	35.29*
	50%	0%	14.75*	24.10*	25.69*	34.72*
	10%	10%	15.10*	24.85*	27.21*	39.72*
	25%	25%	15.88*	25.77*	27.73*	40.52*
	50%	50%	14.72	26.63*	27.82*	40.69*
	100%	100%	15.31*	26.66*	27.51*	40.29*
VRBot-Bert#	50%	50%	15.80	26.70*	28.21	41.18*
	100%	100%	16.11	27.60*	28.80	42.08

5.2 Ablation study

As shown in Table 5, all components in VRBot contribute to its performance. On KaMed, the performance of VRBot\S and VRBot\A

Table 4: Human evaluation on the KaMed and MedDG datasets (with 25% annotations).

Model	# KaMed			Model	# MedDG		
	Flu	KC	EQ		Flu	KC	EQ
SEDST	3.52	1.88	1.81	DAMD	3.77	2.62	2.49
PostKS	3.20	1.77	1.67	MOSS	3.76	2.88	2.59
VRBot	4.21	2.96	2.69	VRBot	4.00	3.26	2.99
κ	0.54	0.56	0.49	κ	0.45	0.47	0.48

drop by 42.27% and 17.64% in terms of mi-F1 respectively. On MedDialog, VRBot\S and VRBot\A drop by 50.54% and 44.76% respectively, which means that states and actions are equally important, modeling only one of them is far from enough. The performance of VRBot\C drops sharply in terms of all metrics; it drops by 22.32% and 58.85% in terms of mi-F1 on KaMed and MedDialog, respectively. VRBot\G drops a little, 3.66% and 1.34% in terms of mi-F1 on KaMed and MedDialog.

The context-reasoning detector is able to leverage the raw dialogue to improve the reasoning ability, whereas the graph-reasoning detector can only use prior knowledge in the knowledge base. In terms of EA and EG, VRBot\C outperforms VRBot\A by 18.12% and 16.98% on KaMed, 28.03% and 26.27% on MedDialog, despite the fact that the mi-F1 score is close to VRBot\A. Hence, VRBot\C benefits from the rich semantics of external knowledge though the entity name in the knowledge graph does not strictly match the dialogue corpus. Without the 2-stage collapsed inference training trick (that is, VRBot\2s), the mi-F1 score decreases by 18.22% and 18.37% on KaMed and MedDialog, respectively.

5.3 Impact of $|S|$ and $|A|$

The length of state and action text span are set to fixed integers $|S|$ and $|A|$ respectively, as they could not be inferred in unsupervised learning. We conduct experiments on MedDialog by setting $|S|$ to values in $\{4, 6, 8, 10, 12\}$ while fixing $|A|$ to 3, and selecting $|A|$ from $\{1, 2, 3, 4, 5\}$ while fixing $|S|$ to 10, to see the effects of $|S|$ and $|A|$. The results are shown in Fig. 5. Focusing on the left part, we see that mi-P decreases, while mi-R and mi-F1 increase as the state text span length grows. As $|S|$ increases from 4 to 10, the mi-R and mi-F1 achieve 12.79% and 4.92% improvements, while mi-P decreases by

Table 5: Ablation study: A comparison of different variations by masking out specific sub-module.

Model	# KaMed			# MedDialog		
	mi-F1	EA	EG	mi-F1	EA	EG
VRBot\S	15.14	30.01	29.64	12.03	21.48	22.05
VRBot\A	18.31	31.67	31.67	12.51	21.26	22.53
VRBot\G	20.78	42.02	41.84	17.87	33.09	35.41
VRBot\C	17.61	37.41	37.05	11.40	27.22	28.45
VRBot\2s	18.22	38.09	37.91	15.30	28.93	30.82
VRBot	21.54	42.37	42.33	18.11	34.51	36.79

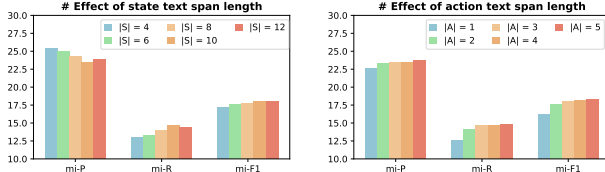


Figure 5: The effect of text span length on MedDialog

6.53%. On the right side, we see a tendency for all metrics to increase as $|A|$ increases, and the upward trend gradually slows down. As $|A|$ increases from 1 to 3, VRBot achieves 3.71%, 16.81% and 11.72% improvements in terms of mi-P, mi-R and mi-F1. The recall score rises a lot as a longer action text spans are able to present more information in the reply. As $|A|$ further increases from 3 to 5, the improvements are relatively small, i.e., 1.04% in terms of mi-F1. We have qualitatively similar findings on the KaMed dataset, which we omit due to space limitations.

5.4 Explainability comparison

To explicitly assess the explainability of VRBot’s results, we calculate MEP and MER scores of action text spans in KaMed. We take a random sample of 50 dialogues from KaMed and manually compare the explainability performance of VRBot and PostKS. The results are listed in Table 6. We observe that VRBot outperforms PostKS by a large margin in terms of MEP and MER; our user study also shows that VRBot achieves a 44% win rate. This confirms that VRBot can provide more interpretable results in responses and action text spans.

5.5 Case study

We randomly sample an example from the KaMed test set to compare the performance of VRBot, SEDST and PostKS in Tab. 7. The dialogue occurs in the ear-nose-throat department and concerns the treatment of ‘allergic rhinitis’. In the 3rd round we see that SEDST and VRBot can both generate a state text span (i.e., S_3 in Tab. 7) to model the patient state. VRBot tracks patient disease and symptoms ‘allergic rhinitis, stuffy nose, sneezing’, then prescribes the correct drugs ‘Nasonex’ and ‘Montelukast’ to meet the patient requirements (it is correct though does not match the gold response); We see a reasoning path ‘allergic rhinitis \rightarrow treated_by \rightarrow Montelukast (0.09)’ in the graph, where 0.09 indicates the copying weight of ‘Montelukast’ in the graph reasoning detector. SEDST and PostKS both fail to generate an accurate and interpretable response; this confirms the importance of simultaneously modeling patient states and physician actions. VRBot is able to generate interpretable responses with explicit text spans and reasoning paths.

Table 6: Explainability comparison of VRBot and PostKS.

Model	MEP	MER	Win Rate (User Study)
PostKS	30.19	56.48	16.00
VRBot	44.22	82.61	44.00

Table 7: One case extracted from KaMed, ✗ and ✓ denote that the response is incorrect and excellent, respectively.

U_1 :	Is it allergic rhinitis (female, 19 years old)? My nose is very itchy and runny after running.
R_1 :	How long has it been? Did you sneeze? Stuffy nose?
U_2 :	Um. I usually feel the nose be a little uncomfortable, often dry nose.
R_2 :	You can use Budesonide Nasal Spray.
U_3 :	Do you have any other suggestions for my symptoms?
Golden :	Flushed your nasal cavity with physiological seawater, take loratadine tablets.
S_3 :	allergic rhinitis, stuffy nose, sneezing, Budesonide Nasal Spray
SEDST:	Spray your nose with Budesonide Nasal Spray. ✗
A_3 :	allergic rhinitis
PostKS:	Your symptom is caused by allergic rhinitis, suggest you go to the hospital to check the nose. ✗
S_3 :	allergic rhinitis, stuffy nose, sneezing
C_n^{local} :	allergic rhinitis \rightarrow treated_by \rightarrow Montelukast (0.09) ; allergic rhinitis \rightarrow treated_by \rightarrow Cetirizine (0.04); allergic rhinitis \rightarrow treated_by \rightarrow Dexamethasone (0.02)
A_3 :	prescribe medicine, Nasonex, Montelukast
VRBot:	Spray your nose with Nasonex , take Montelukast Sodium Chewable Tablets. ✓

6 CONCLUSIONS

In this paper, we focus on medical dialogue response generation with a large-scale unlabeled corpus. We propose a generative model named VRBot, which uses latent variables to model unobserved patient state and physician actions. We derive the ELBO for VRBot and propose a 2-stage collapsed inference training trick that decomposes the ELBO into two learning objectives. Extensive experiments on three medical dialogue datasets show that VRBot achieves state-of-the-art performance on both unsupervised and semi-supervised learning. Furthermore, in a fully-supervised setting, VRBot-Bert which is a variation of VRBot augmented by Bert achieves the best results compared to all the baselines. Analysis also confirms that VRBot is able to generate interpretable results.

VRBot proves the value of having a large-scale unlabeled medical corpus. It can be also applied to other task-oriented dialogue systems with few annotated data. As to our future work, we aim to leverage the labeled data of a single hospital department to improve the MDG performance on other departments without labeled data by transfer learning or zero-shot learning.

REPRODUCIBILITY

Our code and dataset are available at <https://github.com/lddsdu/VRBot>.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (61902219, 61972234, 62072279), the National Key R&D Program of China with grant No. 2020YFB1406704, the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), Shandong University multidisciplinary research and innovation team of young scholars (No. 2020QNQT017), the Tencent WeChat Rhino-Bird Focused Research Program (JR-WXG-2021411), and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, <https://hybrid-intelligence-centre.nl>. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.
- [2] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational Graph Attention Networks. *arXiv preprint arXiv:1904.05811* (2019).
- [3] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A Survey on Dialogue Systems: Recent Advances and New Frontiers. In *SIGKDD*. 25–35.
- [4] Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. 2018. Hierarchical variational memory network for dialogue generation. In *WWW*. 1653–1662.
- [5] Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, Penghui Zhu, and Pengtao Xie. 2020. MedDialog: A Large-scale Medical Dialogue Dataset. In *EMNLP*. 9241–9250.
- [6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. 103–111.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2019), 4171–4186.
- [8] Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019. Extracting Symptoms and their Status from Clinical Conversations. In *ACL*. 915–925.
- [9] Matt W Gardner and SR Dorling. 1998. Artificial Neural Networks (The Multilayer Perceptron)—A Review of Applications in the Atmospheric Sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.
- [10] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A Knowledge-grounded Neural Conversation Model. In *AAAI*. 5110–5117.
- [11] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-sequence Learning. (2016), 1631–1640.
- [12] Matthew Henderson, Blaise Thomson, and Steve Young. 2013. Deep Neural Network Approach for the Dialog State Tracking Challenge. In *SIGDIAL*. 467–471.
- [13] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-oriented Dialogue. In *NeurIPS*. 20179–20191.
- [14] Kai Hua, Zhiyuan Feng, Chongyang Tao, Rui Yan, and Lu Zhang. 2020. Learning to Detect Relevant Contexts and Knowledge for Response Selection in Retrieval-Based Dialogue Systems. In *CIKM*. 525–534.
- [15] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding based Question Answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 105–113.
- [16] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).
- [17] Xisen Jin, Wenqiang Lei, Zhaochun Ren, Hongshen Chen, Shangsong Liang, Yihong Zhao, and Dawei Yin. 2018. Explicit State Tracking with Semi-supervision for Neural Dialogue Generation. In *CIKM*. 1403–1412.
- [18] Byeonchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *ICLR*.
- [19] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
- [20] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.
- [21] Sungjin Lee. 2013. Structured Discriminative Model for Dialog State Tracking. In *SIGDIAL*. 442–451.
- [22] Sungjin Lee and Maxine Eskenazi. 2013. Recipe for Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description. In *SIGDIAL*. 414–422.
- [23] Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2020. Conversational Recommendation: Formulation, Methods, and Evaluation. In *SIGIR*. 2425–2428.
- [24] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-sequence Architectures. In *ACL*. 1437–1447.
- [25] Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the Role of Schema Linking in Text-to-SQL. In *EMNLP*. 6943–6954.
- [26] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. In *SIGKDD*. 2073–2083.
- [27] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL*. 110–119.
- [28] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. *arXiv preprint arXiv:1902.04911* (2019).
- [29] Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. Moss: End-to-end Dialog System Framework with Modular Supervision. In *AAAI*. 8327–8335.
- [30] Lizi Liao, Yunshan Ma, Wenqiang Lei, and Tat-Seng Chua. 2020. Rethinking Dialogue State Tracking with Reasoning. *arXiv preprint arXiv:2005.13129* (2020).
- [31] Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*. 74–81.
- [32] Xinzhu Lin, Xiahui He, Qin Chen, Huaixiao Tou, Zhongyu Wei, and Ting Chen. 2019. Enhancing Dialogue Symptom Diagnosis with Global Attention and Symptom Graph. In *EMNLP-IJCNLP*. 5036–5045.
- [33] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How to Evaluate your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *EMNLP*. 2122–2132.
- [34] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to Evaluate your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *EMNLP*. 2122–2132.
- [35] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge Diffusion for Neural Dialogue Generation. In *ACL*. 1489–1498.
- [36] Wenge Liu, Jianheng Tang, Jinghui Qin, Lin Xu, Zhen Li, and Xiaodan Liang. 2020. MedDG: A Large-scale Medical Conversation Dataset for Building Medical Dialogue System. *arXiv preprint arXiv:2010.07497* (2020).
- [37] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. Knowledge Aware Conversation Generation with Reasoning on Augmented Graph. *arXiv preprint arXiv:1903.10245* (2019).
- [38] Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A Toolkit for Multi-domain Chinese Word Segmentation. *arXiv preprint arXiv:1906.11455* (2019).
- [39] Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured Fusion Networks for Dialog. In *SIGDIAL*. 165–177.
- [40] Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and Maarten de Rijke. 2020. Dukenet: A Dual Knowledge Interaction Network for Knowledge-grounded Conversation. In *SIGIR*. 1151–1160.
- [41] Nikola Mrksić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural Belief Tracker: Data-Driven Dialogue State Tracking. (2017), 1777–1788.
- [42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.
- [43] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *SIGIR*. 539–548.
- [44] Iulian V Serban, Alessandro Sordani, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-to-end Dialogue Systems using Generative Hierarchical Neural Network Models. In *AAAI*. 3776–3783.
- [45] Iulian Vlad Serban, Alessandro Sordani, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. A Hierarchical Latent Variable Encoder-decoder Model for Generating Dialogues. In *AAAI*. 3295–3301.
- [46] Xiaoming Shi, Haifeng Hu, Wanxiang Che, Zhongqian Sun, Ting Liu, and Junzhou Huang. 2020. Understanding Medical Conversations with Scattered Keyword Attention and Weak Supervision from Responses. In *AAAI*. 8838–8845.
- [47] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*. 3104–3112.
- [48] Maartje ter Hoeve, Robert Sim, Elhaz Nouri, Adam Fournay, Maarten de Rijke, and Ryen White. 2020. Conversations with Documents: An Exploration of Document-Centered Assistance. In *SIGIR*. 43–52.
- [49] Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. DyKgChat: Benchmarking Dialogue Generation Grounding on Dynamic Knowledge Graphs. In *EMNLP-IJCNLP*. 1855–1865.
- [50] Svitlana Vakulenko, Evangelos Kanoulas, and Maarten de Rijke. 2020. An Analysis of Mixed Initiative and Collaboration in Information-Seeking Dialogues. In *SIGIR*. 2085–2088.
- [51] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *CIKM*. 417–426.
- [52] Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat More: Deepening and Widening the Chatting Topic via a Deep Model. In *SIGIR*. 255–264.
- [53] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge Graph Attention Network for Recommendation. In *SIGKDD*. 950–958.
- [54] Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuan-Jing Huang, Kam-Fai Wong, and Xiang Dai. 2018. Task-oriented Dialogue System for Automatic Diagnosis. In *ACL*. 201–207.
- [55] Tsung-Hsien Wen, David Vandyke, Nikola Mrksić, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *EACL*. 438–449.
- [56] Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2021. Alternating Recurrent Dialog Model with Large-scale Pre-trained Language Models. In *EACL*. 1292–1301.
- [57] Yuan Xia, Jingbo Zhou, Zhenhui Shi, Chao Lu, and Haifeng Huang. 2020. Generative Adversarial Regularized Mutual Information Policy Gradient Framework for Automatic Diagnosis. In *AAAI*. 1062–1069.

- [58] Jun Xu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Conversational Graph Grounded Policy Learning for Open-domain Conversation Generation. In *ACL*. 1835–1845.
- [59] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019. End-to-end Knowledge-routed Relational Dialogue System for Automatic Diagnosis. In *AAAI*. 7346–7353.
- [60] Wenmian Yang, Guangtao Zeng, Bowen Tan, Zeqian Ju, Subrato Chakravorty, Xuehai He, Shu Chen, Xingyi Yang, Qingyang Wu, Zhou Yu, Eric Xing, and Pengtao Xie. 2020. On the Generation of Medical Dialogues for COVID-19. *arXiv preprint arXiv: 2005.05442* (2020).
- [61] Yan Zeng and Jian-Yun Nie. 2020. Multi-Domain Dialogue State Tracking–A Purely Transformer-Based Generative Approach. *arXiv preprint arXiv: 2010.14061* (2020).
- [62] Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. 2020. Summarizing and Exploring Tabular Data in Conversational Search. In *SIGIR*. 1537–1540.
- [63] Yongfeng Zhang and Xu Chen. 2018. Explainable Recommendation: A Survey and New Perspectives. *arXiv preprint arXiv: 1804.11192* (2018).
- [64] Yuanzhe Zhang, Zhongtao Jiang, Tao Zhang, Shiwan Liu, Jiarun Cao, Kang Liu, Shengping Liu, and Jun Zhao. 2020. MIE: A Medical Information Extractor towards Medical Dialogues. In *ACL*. 6460–6469.
- [65] Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020. A Probabilistic End-To-End Task-Oriented Dialog Model with Latent Belief States towards Semi-Supervised Learning. In *EMNLP*. 9207–9219.
- [66] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. In *AAAI*. 9604–9611.
- [67] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In *IJCAL*. 4623–4629.
- [68] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible End-to-end Dialogue System for Knowledge Grounded Conversation. *arXiv preprint arXiv:1709.04264* (2017).