Semi-Supervised Variational Reasoning for Medical Dialogue Generation

Dongdong Li1∗, Zhaochun Ren1†, Pengjie Ren1, Zhumin Chen1, Miao Fan2, Jun Ma1, Maarten de Rijke3,4
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.

CSC 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

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...
Pyrazinamide

The combination of four drugs, Isoniazid, Ethambutol, Rifampicin, and Atropine, is used to treat tuberculosis.

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
TDGs 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 KG and unstructured KG. 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 KG, 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., $d = \{U_1, R_1, U_2, R_2, ... , U_T, R_T\}$, 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 $U_t = (U_{t,1}, U_{t,2}, ... , U_{t,|U_t|})$ as a sequence of words. The full vocabulary is defined as $V$. $K$ 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 entities (i.e., two triples will be linked if they share overlapping entities) in $K$. We assume that each entity is categorized into a set of entity types, i.e., $E_{type} = \{disease, symptoms, medicines, treatments\}$.

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

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

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., $U_1, R_1, ... , U_{t-1}, U_t$). 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:

$$\prod_{t=1}^{T} \sum_{S_t, A_t} p_{\theta_y}(R_t|R_{t-1}, U_t, S_t, A_t) \cdot p_{\theta_x}(S_t) \cdot p_{\theta_a}(A_t).$$

where $p_{\theta_y}(R_t|R_{t-1}, U_t, S_t, A_t)$ is derived using a response generator, and $p_{\theta_x}(S_t)$ 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 $p_{\theta_x}(S_t)$ and $p_{\theta_a}(A_t)$ as:

$$p_{\theta_x}(S_t) \approx p_{\theta_a}(S_t|S_{t-1}, R_{t-1}, U_t) \text{ (prior state tracker)},$$

$$p_{\theta_a}(A_t) \approx p_{\theta_a}(A_t|S_t, R_{t-1}, U_t, G^{global}) \text{ (prior policy network)},$$

![Figure 2: The graphical representation of VRBot. Shaded nodes represent observed variables.](image-url)
where $\theta_s$ and $\theta_a$ 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 $p_{\theta_R}(R_t|R_{t-1}, U_t, S_t, A_t)$, with parameters $\theta_R$.

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:

$$ q_{\phi_s}(S_t) \approx q_{\phi_s}(S_t|R_{t-1}, R_t, U_t, R_t) \text{ (inference state tracker)} $$

$$ q_{\phi_a}(A_t) \approx q_{\phi_a}(A_t|S_t, R_{t-1}, U_t, R_t) \text{ (inference policy network)} $$  

where $\phi_s$ and $\phi_a$ 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:

$$ \log p_{\theta}(R_t|R_{t-1}, U_t, G_{global}) \geq \mathbb{E}_{q_{\phi_s}(S_t)} \left[ \mathbb{E}_{q_{\phi_a}(A_t)} \left[ \log p_{\theta}(R_t|R_{t-1}, U_t, S_t, A_t) \right] \right] $$

$$ -\mathbb{KL}(q_{\phi_s}(S_t)||p_{\theta_s}(S_t)) - \mathbb{KL}(q_{\phi_a}(A_t)||p_{\theta_a}(A_t)) \right] $$

$$ = -L_{\text{elbo}} $$

where $\mathbb{E}()$ 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 $S_{t-1}^p$, 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 $S_t^p$ 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 $S_t^p$, respectively, and draw $A_t^p$ from $q_{\phi_a}(A_t)$. Finally, $p_{\theta_R}(R_t|\cdot)$ generates $R_t$ depending on $S_t$ and $A_t$. 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 $H_t = (h_{t,1}, \ldots, h_{t,|U_t|+|R_{t-1}|})$ using a bi-directional Gated Recurrent Unit (GRU) [6]:

$$ H_t = \text{BiGRU}(h_{t-1}^f, e_{t-1}^R; e_{t-1}^L, e_{t-1}^R, \ldots, e_{t-1}^R, U_{t-1}, U_t) $$  

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

Initializing from the hidden representation $h_{t-1}^f$ 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., $h_t^f$.

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 $S_{t-1}^p$ using a GRU encoder to get $h_t^f$ during the encoding procedure. We then incorporate $h_t^f$ with $h_t^f$ 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 $b_{t}^{SP} = W_t^q [h_t^f; h_t^{SP}]$ as the initial hidden representation of the decoder, where $W_t^q$ 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 $b_{t}^{SP}$ given previous token embedding $e_{t-i}^{SP}$, next projects $b_{t}^{SP}$ into the patient state space. We set $S_t$’s length to $|S_t|$ and the prior distribution over $S_t$ is calculated as:

$$ p_{\theta_s}(S_t) = \sum_{i=1}^{|S_t|} \text{softmax}(\text{MLP}(b_{t}^{SP})), $$

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., $h_t^f$. The GRU decoder is initialized as $b_{t}^{SP} = W_t^q [h_t^f; h_{t-1}^f; h_t^f]$, where $W_t^q$ is a learnable parameter, and it outputs $b_{t}^{SP}$ at the i-th decoding step. Accordingly, we write the approximate posterior distribution as:

$$ q_{\phi_s}(S_t) = \sum_{i=1}^{|S_t|} \text{softmax}(\text{MLP}(b_{t}^{SP})). $$

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 $A_t^f$ and a list of explicit keywords $A_t^k$, i.e., $A_t = (A_t^f, A_t^k)$. Here we set the length of $A_t^k$ to $|A_t^k|$.
As for the prior policy network, at the beginning of the encoding procedure, we encode \( S_f \) to a vector \( h^S_f \) 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 \( G^\text{global} \) is large (in the number of entities), we extract a sub-graph \( G^n \) from \( G^\text{global} \) via a knowledge base retrieval operation \( \text{qsub} \), where we regard each entity in \( S_f \) as seed nodes during \( \text{qsub} \), where we regard each entity in \( S_f \) as seed nodes during \( \text{qsub} \). Starting from \( S_f \), we extract all the accessible nodes and edges in \( G^n \) within \( n \) hops to get the sub-graph \( G^n \) [51]. Besides, we link all the entities appear in \( S_f \) to ensure \( G^n \) 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 \( G = [g_1, g_2, \ldots, g_X] \), where \( g_x, (1 \leq x \leq X) \) is the embedding of node \( x \). We use RGAT to denote this operation, so we have: \( G^n = \text{RGAT}(G^n) \).

To decode outputs, we infer \( A^c_i \) and \( A^s_i \) sequentially. We devise an action classifier to infer \( A^c_i \). Following [1], we compute an attention vector \( q_i \) over \( G^n \) with \( h^c_i \) as the query. Sequentially, the action classifier incorporates \( q_i \), and classifies physician action into four categories, i.e., ask symptoms, diagnosis, prescribe medicine and chitchat, as follows:

\[
P_{\phi_{ac}}(A^c_i) = \text{softmax}(W_p^c(h^{SP}_i; h^c_i; q_i)),
\]

where \( W_p^c \) is a learnable parameter. Then we draw an action category \( A^c_i \) by sampling from \( p_{\phi_{ac}}(A^c_i) \).

\( A^s_i \) 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 present the hidden representation of the decoder to the action space at each decoding step. The decoder is initialized as \( b^k_{t,0} = W_k^c[h^c_i; h^c_t; e^{k-\text{SP}}_t] \), where \( e^{k-\text{SP}}_t \) is the embedding of \( A^c_{t-1} \). At the \( t \)-th decoding step, the decoder outputs \( b^k_{t,i} \).

The context-reasoning detector and the graph-reasoning detector together infer \( A^c_i \) with \( b^k_{t,i} \).

Learning from the raw context and state, the context-reasoning detector infers the prior distribution over \( A^c_{t,i} \) using an MLP as follows:

\[
p_{\phi_{ac}}(A^c_{t,i}) = \frac{1}{z_A} \exp(\text{MLP}((h^c_t; h^c_t; b^k_{t,i}))),
\]

where \( z_A \) is the normalization term shared with the graph-reasoning detector. The graph-reasoning detector considers to copy entities from \( G^n \):

\[
p_{\phi_{ad}}(A^s_{t,i}) = \frac{1}{z_A} \cdot \exp(W_g[h^c_t; b^k_{t,i}; g_j]),
\]

where \( W_g \) is a learnable parameter matrix, \( e_j \) is the \( j \)-th entity in \( G^n \), \( g_j \) is the \( j \)-th entry embedding of \( G^n \), \( l(e_j, A^s_{t,i}) \) equals 1 if \( e_j = A^s_{t,i} \) otherwise 0. Eventually, we calculate the prior distribution over \( A_t \) as follows:

\[
p_{\phi}(A_t) = p_{\phi_{ac}}(A^c_t) \cdot \prod_{i=1}^{\lfloor |A_t|/2 \rfloor} [p_{\phi_{ac}}(A^c_{t,i}) + p_{\phi_{ad}}(A^s_{t,i})].
\]

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 \( h^R_t \), \( S_f \) to \( h^{SP}_f \) respectively. Then we get the action category approximate posterior distribution as follows:

\[
q_{\phi_{ac}}(A^c_t) = \text{softmax}(W_p^c [h^c_t; h^{SP}_f; h^R_t]).
\]

Thereafter, we draw \( A^c_t \) via sampling from \( q_{\phi_{ac}}(A^c_t) \). To reinforce the effect of information of \( R_t \), we only use the context-reasoning detector to approximate the posterior distribution of \( A^c_t \). The decoder is initialized as \( b^k_{t,0} = W_k^c[h^c_t; h^{SP}_f; e^{c-\text{SP}}_t, h^R_t] \), where \( e^{c-\text{SP}}_t \) is the embedding of \( A^c_{t-1} \). \( W_k^c \) reflects a learnable parameter matrix. At the \( t \)-th decoding step, the decoder outputs \( b^k_{t,i} \) so we have the approximate posterior distribution over the \( t \)-th action keyword:

\[
q_{\phi_{ad}}(A^s_{t,i}) = \text{softmax}(\text{MLP}([h^c_t; h^{SP}_f; b^k_{t,i}])).
\]

Eventually, we get the approximate posterior distribution of \( A_t \):

\[
q_{\phi}(A_t) = q_{\phi_{ac}}(A^c_t) \cdot \prod_{i=1}^{\lfloor |A_t|/2 \rfloor} q_{\phi_{ad}}(A^s_{t,i}).
\]

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

### 3.5 Response generator

At the first stage during the response generation, we use a GRU encoder to encode \( S_f \) into \( S_f^n \) which is a word-level embedding matrix of \( S_f \). Each column vector in \( S_f^n \) reflects an embedding vector of the corresponding word in \( S_f \). In the same manner, \( A^c_{t-1} \) is encoded to \( A^c_{t-1} \). As mentioned in Sec. 3.3 and 3.4, we also calculate the holistic embedding \( h^{SP}_t \) and \( h^{c_{i,j}}_t \) from \( S_f^n \) and \( A^c_{t-1} \), respectively. The response decoder with a GRU cell takes \( b^k_{t,0} = W_k^c[h^c_t; h^{SP}_t; e^{c_{i,j}}_t, h^R_t] \) as the initial hidden state.

At the \( t \)-th decoding step, the output \( b^k_{t,i} \) from the \( t \)-th step attentively reads the context representation \( H_t \) to get \( b^k_{t,i} \).

Meanwhile, \( b^k_{t,i} \) attentively reads \( S_f^n \) and \( A^c_{t-1} \) to get \( b^k_{t,i} \) and \( b^k_{t,i} \) respectively. Subsequently, \( [b^k_{t,i}; b^k_{t,i}; e^{c_{i,j}}_{t-1}] \) are fed into the decoder GRU cell to output \( b^k_{t,j} \) where \( e^{c_{i,j}}_{t-1} \) is the embedding of \((i-1)\)-st word in \( R_t \). The probability of generating \( R_t \) is formulated as a sum of the generative probability and a copy term:

\[
p_{\phi_{ad}}(R_t) = p_{\phi_{ad}}(R_{t,i}) + p_{\phi_{ad}}(R_{t,i}).
\]

\[
p_{\phi_{ad}}(R_t) = \frac{1}{z_R} \exp(\text{MLP}(h^R_t)),
\]

\[
p_{\phi_{ad}}(R_t) = \frac{1}{z_R} \sum_{j \neq w} \exp(h^W_j \cdot h^R_t),
\]

where \( p_{\phi_{ad}}(R_{t,i}) \) is the generative probability, \( p_{\phi_{ad}}(R_{t,i}) \) is the copy term, \( z_R \) is the normalization term shared with \( p_{\phi_{ad}}(R_{t,i}) \). We write \( W \) for a concatenation sequence of \( R_{t-1}, U_t, S_{t-1} \) and \( A_{t-1} \), where \( W_j \) is the \( j \)-th word in \( W \), and \( h^W_j \) is the \( j \)-th vector in \( \{H_i; S_f^n; A^c_{t-1}\} \).
We seek to answer the following research questions: (RQ1) How does VRBot perform on medical dialogue generation? (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, 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, 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-automated 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 \(L_{\text{joint}}\) 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 models to perform semi-supervised training.

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 \(p_\theta(A_t) \cdot p_\theta(S_t)\) is hard to optimize as \(p_\theta(A_t)\) is easily misled with incorrect sampling results of \(\bar{S}_t\) from \(p_\theta(S_t)\). 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 \(p_\theta(S_t)\) to \(q_{\phi_S}(S_t)\) to derive the ELBO (labeled by \(\bullet\) in Fig. 4):
We recruit three professional annotators from a third-party hospital to evaluate the responses generated by all models 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, and mi-F1. In terms of ma-R, mi-R, ma-F1, and mi-F1, VRBot outperforms MOSS by 11.90% and 10.36% 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\text{run} 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 ≤ 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.

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, and mi-F1. In terms of ma-R, mi-R, ma-F1, and mi-F1, VRBot outperforms MOSS by 11.90% and 10.36% 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\text{run} 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 ≤ 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.

1https://code.ihub.org.cn/projects/1775/repository/mindspore_pretrain_bert
As shown in Table 5, all components in VRBot contribute to its performance. On KaMed, the performance of VRBot\$ and VRBot\{A decrease by 42.27% and 17.64% in terms of mi-F1 respectively. On MedDialog, VRBot\$ 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 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 fact terms. 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.

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
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 comparsion

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); allergic rhinitis $\rightarrow$ treated_by $\rightarrow$ Cetirizine (0.04); allergic rhinitis $\rightarrow$ treated_by $\rightarrow$ Dexamethasone (0.02)’.

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-202114), the Hybrid Intelligence Center, and 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.
[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, Qinxian 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, Kristtian 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, Jiurun Cao, Kang Liu, Shengping Liu, and Jun Zhao. 2020. MIE: A Medical Information Extractor towards Medical Dialogues. In ACL. 6466–6469.

[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 IJCAI. 4623–4629.

[68] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezhe Peng, and Qiang Yang. 2017. Flexible End-to-end Dialogue System for Knowledge Grounded Conversation. arXiv preprint arXiv:1709.04264 (2017).