Generative Adversarial Regularized Mutual Information Policy Gradient Framework for Automatic Diagnosis

Yuan Xia[#], Jingbo Zhou[†*], Zhenhui Shi[#], Chao Lu[#], Haifeng Huang[#]
[#]Baidu Inc, Beijing, China [†]Business Intelligence Lab, Baidu Research
[†]National Engineering Laboratory of Deep Learning Technology and Application, China
{xiayuan, zhoujingbo, shizhenhui, luchao, huanghaifeng}@baidu.com

Abstract

Automatic diagnosis systems have attracted increasing attention in recent years. The reinforcement learning (RL) is an attractive technique for building an automatic diagnosis system due to its advantages for handling sequential decision making problem. However, the RL method still cannot achieve good enough prediction accuracy. In this paper, we propose a Generative Adversarial regularized Mutual information Policy gradient framework (GAMP) for automatic diagnosis which aims to make a diagnosis rapidly and accurately. We first propose a new policy gradient framework based on the Generative Adversarial Network (GAN) to optimize the RL model for automatic diagnosis. In our framework, we take the generator of GAN as a policy network, and also use the discriminator of GAN as a part of the reward function. This generative adversarial regularized policy gradient framework can try to avoid generating randomized trials of symptom inquires deviated from the common diagnosis paradigm. In addition, we add mutual information to enhance the reward function to encourage the model to select the most discriminative symptoms to make a diagnosis. Experiment evaluations on two public datasets show that our method beats the state-of-art methods, not only can achieve higher diagnosis accuracy, but also can use a smaller number of inquires to make diagnosis decision.

Introduction

Automatic diagnosis is one of the most important artificial intelligence applications in healthcare. An automatic diagnosis system usually converses with patients a series of questions about their symptoms beyond their self-reports and then attempts to predict potential diseases. The automatic diagnosis system has a great potential to simplify the diagnostic procedure, reduce the cost of collecting patient information, and help make a better and more efficient decision making (Tang et al. 2016; Liu et al. 2017; Chen et al. 2019).

In recent years, researchers are increasingly interested in modeling the automatic diagnosis problem by reinforcement learning (RL). The process of automatic diagnosis can be considered as a sequence of inquiries from doctors and answers from patients. Meanwhile, a distinctive feature of RL is to tackle with sequential decision making problems with feedback. Therefore, RL is popularly considered as a suitable candidate for developing powerful solutions for automatic diagnosis (Yu, Liu, and Nemati 2019).

However, there are still several challenges for RL to solve automatic diagnosis. First of all, with the limited size of diagnosis data, the RL tends to generate randomized trials without considering the correlations among symptoms and diseases from the common diagnosis paradigm. In real-life scenarios, the doctors always carefully choose relevant questions to asks the patients with a logic of medical diagnosis. The RL requires a large amount of data to learn such latent knowledge, while the size of diagnosis data are much small due to the cost to collect the data and the privacy concern of patients. Second, a sophisticated method to set the reward function is necessary. Though existing works have mentioned that rewards are crucial for policy learning in RL (Xu et al. 2019), there still no good solution to set the reward function. For example, in (Kao, Tang, and Chang 2018), the positive reward is set as $+1$, and negative reward as 0, whereas in (Xu et al. 2019), the positive reward is set as $+44$ and the negative reward as -22. There is no insightful intuition or solution for setting the reward values in different application contexts.

In this paper, we introduce a dialogue system for automatic diagnosis, and propose a novel Generative Adversarial regularized Mutual information Policy gradient framework (named GAMP for short) for automatic diagnosis in our system. Figure 1 illustrates our system architecture, which contains Dialogue Agent (DA), User Simulator (US), Natural Language Understanding (NLU) and Natural Language Generation (NLG). Additionally, we have a Dialogue State Tracker (DST) to track the state of the user and agent. The NLU extracts medical entities and key question from the input text, and NLG can generate the dialogue question to patients. The Dialogue Agent implements GAMP framework, as shown in Figure 2.

The novelty of GAMP framework lays on the integration of the Generative Adversarial Network (GAN) (Goodfellow et al. 2014) with the RL model. We propose to train an RL

*Jingbo Zhou is the corresponding author.

model and a GAN simultaneously for automatic diagnosis, with taking the generator of GAN as policy network of RL. The discriminator of GAN (named as evaluation discriminator) can be used to estimate a likelihood that the symptom sequence is "real" (instead of "fake") sequence asked by doctors. We use the evaluation discriminator to design a reward function to guide the optimization of policy network. We call such a new policy learning strategy as generative adversarial regularized policy gradient. The insight of the proposed method is that the doctors usually choose to ask relevant questions to the patient with prior medical knowledge. For example, after asking a question "do you get a headache" with having a "yes" answer, few of doctors will ask "do you have foot pain" since these symptoms almost impossible exist for the same disease. However, the RL tends to generate randomized trials for the symptoms without considering the common diagnosis paradigm. With the limited size of training data, the RL cannot capture such latent and complex medical knowledge. Thus, in our framework, we use the likelihood of the evaluation discriminator to regularize and enhance the policy network, resulting in generating better symptom inquiry sequence.

Second, we propose to use mutual information (MI) to further enhance the reward function to optimize the model. Our observation is that, during the diagnosis process, the doctor usually inquires the most discriminative symptom which can eliminate the uncertainty as much as possible to make a differential diagnosis. In our framework, we devise a mechanism to compute the MI between the current state disease probability distribution and the adjacent next state disease probability based on an inference engine. Then we combine MI into the reward function for the policy learning.

We summary our contribution as follows:

- We propose a new framework (GAMP) for automatic diagnosis. GAMP has two novel techniques, which are generative adversarial regularized policy gradient and mutual information enhanced reward function, to optimize the policy learning with policy gradient.
- We introduce a complete description of our dialogue system, which incorporates the GAMP framework for automatic diagnosis.
- We evaluate our system on two public datasets to demonstrate the superiority of our framework with higher accuracy and less inquires to make a diagnosis decision.

Related Work

Task-oriented dialogue systems are attracting more and more attention in recent years. Sequence-to-sequence models have been used in task-oriented dialogue systems (Sutskever, Vinyals, and Le 2014; Eric and Manning 2017; Lei et al. 2018). Recent studies on text generation on GAN have been a highly active area. SeqGAN (Yu et al. 2017) trains a language model with policy gradients to train the generator to deceive a CNN-based discriminator. In order to acquire a meaningful loss in every token, they do Monte Carlo rollouts during the training. GANs have been applied to dialogue generation (Li et al. 2017) showing improvements in adversarial evaluation and good results with human evaluation compared to a maximum likelihood trained baseline. There are variants of GANs, like Conditional-GAN (Mirza and Osindero 2014), Info-GAN (Chen et al. 2016) and AC-GAN (Odena, Olah, and Shlens 2017), training GANs with additional class or tag information. (Chen et al. 2016) use an variational information maximization technique to optimize GANs.

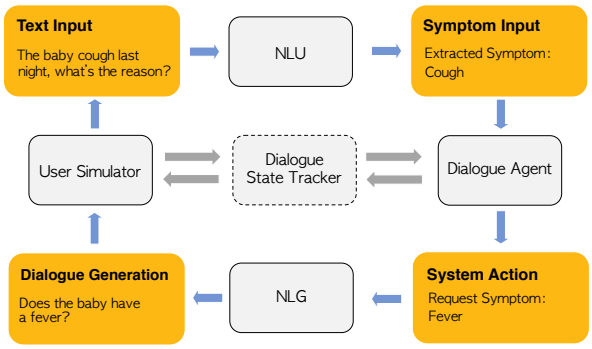

Figure 1: Architecture of Dialogue System for Automatic Diagnosis

There are some works related to our study. Deep reinforcement learning (Mnih et al. 2013; Silver et al. 2016; 2017) has been applied for automatic diagnosis (Tang et al. 2016; Kao, Tang, and Chang 2018). (Peng et al. 2018) proposed reward shaping and feature rebuilding method for fast disease diagnosis. However, their data used is simulated that cannot reflect the situation of the real diagnosis. For the medical dialogue system for automatic diagnosis, (Liu et al. 2018) annotated the first medical dataset for dialogue system and use a Deep Q-network (DQN) to collect additional symptoms via conversation with patients. (Xu et al. 2019) released another medical dataset for the dialogue system and introduce prior knowledge to improve the diagnosis accuracy. However, with fixed reward function, the intuition of reward is unclear, their DQN based methods fail to request the distinguished symptoms and sometimes request unreasonable results.

Proposed Method

The architecture of the proposed dialogue system is illustrated in Figure 1. In this work, we mainly focus on the Dialogue Agent (DA) which is composed of a generator, a discriminator and an inference engine: the generator is used for inquiring the patient with possible symptoms; the discriminator is used for evaluating whether the inquired sequence is authentic or fake; and the inference engine is used for inferring the possible diseases. The user simulator in Figure 1 is normally designed to automatically interact with the Dialogue Agent (Liu et al. 2017). The NLU can recognize user intent and normalize the symptoms from the patient self-report and conversations. The NLU is implemented with a simple Bi-LSTM model. Given the predicted symptom actions, the NLG is used for generating natural language sen-

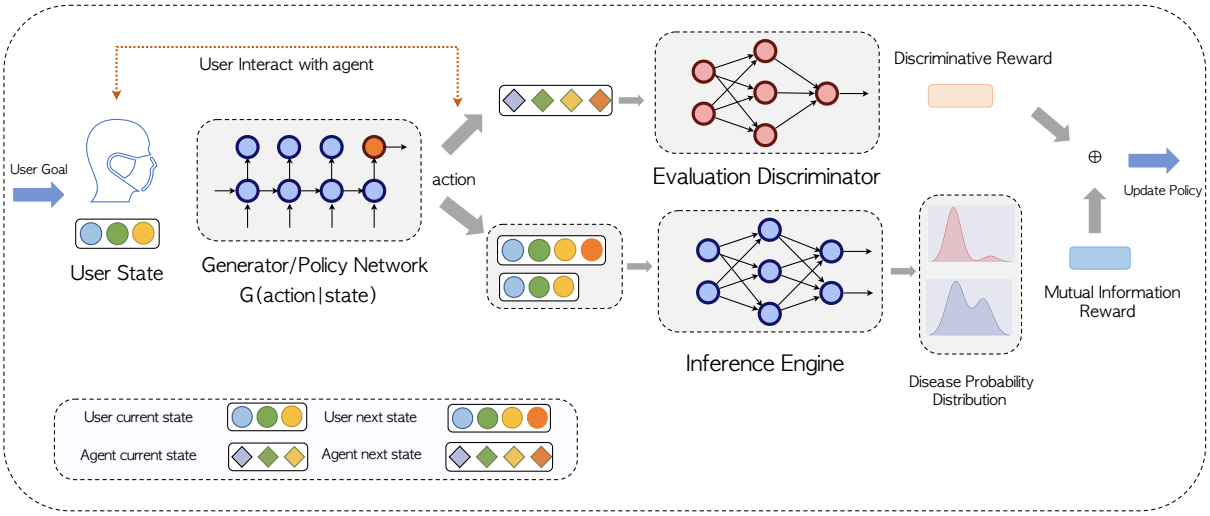

Figure 2: Illustration of the GAMP Framework for Dialogue Agent

tences via templates. The whole dialogue agent system receives the patient self-report, learns to request symptoms to interact with the patient, and makes a diagnosis at the end of the conversation. When the maximum turns T is reached, or the inference engine reaches the threshold τ, or the generator chooses the terminated node (disease node), the dialogue session will terminate.

Suppose we have a medical dataset consisting of patients' symptoms and disease. We have m disease and n symptoms. The agent has four types of actions including inform disease, request symptom, thanks and terminate (Liu et al. 2018). Therefore the agent action space size $\mathcal{A}$ is denoted as $\mathcal{A} = m + n + l$, where l is the size of auxiliary action (e.g. thanks and terminate). User actions are mainly confirmation, denial or uncertainty about the requested symptoms. There are three types of symptom states for a user, which are positive, negative and uncertain, represented by 1, -1 and 0 in user symptom vectors respectively. At each turn, the DST records the dialogue state s_t which contains the previous actions of both user and agent, known symptoms representation and current turn information.

User Simulator

To train our dialogue system, we apply a user simulator to sample user goals from the experimental dataset $\mathcal{S}$ for automatically interacting with the dialogue system. Following (Liu et al. 2018), our user simulator maintains a user goal. A user goal generally consists of four parts: disease tag, explicit symptoms (symptoms extracted from self-reports), implicit symptoms (symptoms extracted from conversations) and request slots. When the agent requests a symptom during the dialogue, the user will take one of the three actions including True (for the positive symptom), False (for the negative symptom), and Not sure (for the unmentioned symptom). At the end of the dialogue session, the user simulator will judge whether the agent makes a correct diagnosis or not. The dialogue process fails if the agent makes wrong diagnosis or the dialogue turn reaches the maximum turn T.

Policy Learning of Dialogue Agent

A framework overview of the dialogue agent system is illustrated in Figure 2. Algorithm 1 shows the summary of our proposed method for automatic diagnosis. First, we pre-train the symptom sequence generator G_θ with the patient self-report and the conversations. The training process is typically a Maximum Likelihood Estimate (MLE) method for language model. Given the symptom sequence extracted from the dialogue, we iteratively predict the next symptom token using an LSTM model. During the training, we predict the next symptom y_t given the input sequence $Y_{1:t-1}$. After the pre-training of generator G_θ, we proceed a roll-out to generate fake symptom sequence. We build a data repository to store the generated symptom sequence, and put fake sequences from the generator into the fake data set.

Second, we pre-train the evaluation discriminator D_ψ with the real symptom sequence and fake sequence. The fake sequence is a combination of the patient self-report and conversations sampled from the generator. The real symptom sequence are sampled from the real medical dialogue training dataset $\mathcal{S}$. The discriminator D_ψ is pre-trained for evaluating the inquired sequence is authentic or not. Meanwhile, we train inference engine D_ϕ to infer possible diseases. The input of D_ϕ are the patient self-report and conversations, the label is the doctor's diagnosis. Note that, as shown in Line 4 of Algorithm 1, once the training of inference engine D_ϕ is finished, D_ϕ will not be updated in our framework. Both the evaluation discriminator and inference engine use the same network architecture, except for the final output layer. The final output layer of discriminator D_ψ is a binary classification, while the inference engine D_ϕ is a softmax layer with cross-entropy loss.

Following (Sutton et al. 2000), the objective of the gen-

Algorithm 1 Generative Adversarial Regularized Mutual Information Policy Gradient

Input: sequence generator G_θ; evaluation discriminator D_ψ and inference engine D_ϕ; a symptom sequence dataset $\mathcal{S}$
Output: optimal policy G_θ^*

Initial G_θ, D_ψ and D_ϕ with random weights θ ,ψ and ϕ.
Pre-train G_θ using MLE on $\mathcal{S}$
Pre-train D_ψ with fake data by G_θ and real data on $\mathcal{S}$
Train D_ϕ using the cross-entropy loss on $\mathcal{S}$
repeat
  for g-steps **do**
    for $t = 1$ to T **do**
      Request symptom $y_t \sim G_\theta$
      Interact with User Simulator
      Compute $R_D = Q_{D_\psi}^{G_\theta}(s = Y_{1:t-1}, a = y_t)$
      Compute $R_M = I(O_t; O_{t+1}|D_\phi)$
      Compute $R_F = (1-\lambda)R_M + \lambda(R_D - \epsilon)$
    end for
    Update the generator parameters via policy gradient Eq.(9)
  end for
  for d-steps **do**
    Sample generated sequence from the fake repository
    Sample real sequence from the real dataset
    Combine the positive and negative samples
    Train discriminator D_ψ for k epochs by Eq.(3)
  end for
until End of epochs

erator (policy) $G_\theta(a_t|s_t)$ is to generate a sequence from the start state s to maximize its expected end reward:

$$J(\theta) = \mathbb{E}[R_T|s, \theta] = \sum_\tau G_\theta(a|s) \cdot Q_{D_\psi}^{G_\theta}(s, a) \tag{1}$$

where R_T is the reward for a symptom sequence, and τ is the trajectory. Note that the reward is from the discriminator D_ψ. $Q_{D_\psi}^{G_\theta}(s, a)$ is the state-action function, which approximates the value when take action a at current state s. The intuition of the objective function for a sequence is that starting from a given initial state (i.e. user self-report in this paper), the goal of the generator G_θ is to generate a sequence which would make the discriminator consider it is real. To estimate the $Q_{D_\psi}^{G_\theta}(s, a)$ function, we use the REINFORCE algorithm (Williams 1992) and consider the estimated probability of evaluation discriminator D_ψ as the reward.

In (Yu et al. 2017), the estimated reward given by the discriminator is computed by the completed sequence. The unobserved sequence $Y_{t+1:T}$ is generated by a roll-out policy with Monte Carlo search method. The drawback of MC is that it requires repeating the sampling process for each prefix of each sequence and is thus significantly time-consuming. Different from the general sequence generation problem, for interactive dialogue system, we not only care about the finished sequence, but also put emphasis on the intermediate feedback. As described in (Li et al. 2017), we directly train a discriminator that is able to assign rewards to both fully and partially observed sequences. We compute the intermediate reward from the partial observed symptom sequence as well as the complete symptom sequence. Thus, we have dicriminator reward R_D:

$$R_D = Q_{D_\psi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = D_\psi(Y_{1:t-1}) \tag{2}$$

Then, once having a set of more realistic generated sequences, we re-train the discriminator D_ψ as follows:

$$\min_\psi -\mathbb{E}_{Y\sim p_{data}}[\log D_\psi(Y)] - \mathbb{E}_{Y\sim G_\theta}[\log(1 - D_\psi(Y)] \tag{3}$$

Mutual Information Regularized Policy Gradient In a real-world process of medical diagnosis, the doctor often inquires the discriminative symptom to make a differential diagnosis. Inspired by this, our generator G_θ is updated to have the ability to inquiry the key symptoms that can best distinguish the diseases which are hard to differentiate. To this end, our work introduces the Mutual Information (MI) to improve the dialogue agent performance.

The entropy measures uncertainty of the events. To reduce the uncertainty step by step, the generator G_θ needs to consider the symptoms which can eliminate the uncertainty, which means can reduce the entropy of disease probability distribution. In information theory, the mutual information between X and Y, $I(X; Y)$, measures the amount of information learned from knowledge of random variable Y about the other random variable X. The mutual information can be expressed as the difference of two entropy terms:

$$I(X; Y) = H(X) - H(X|Y) \tag{4}$$

In our work, we compute the mutual information between the current state disease probability distribution O_{t-1} and adjacent next state disease distribution O_t. The mutual information is computed as follows:

$$I(O_{t-1}; O_t|D_\phi) = H(O_{t-1}|D_\phi) - H(O_t|D_\phi) \tag{5}$$

$$O_{t-1} = D_\phi(Y'_{1:t-1}) \tag{6}$$

where, $H(\cdot)$ is the entropy function, $Y'_{1:t-1}$ is a collection of symptoms that the patient confirms to have (circles shown in Fig. 2), while $Y_{1:t-1}$ is the symptom sequence has been inquired by agent (diamonds shown in Fig. 2). y_t is the next symptom to be asked.

In the process of medical diagnosis, the doctor inquiries the possible symptoms. It normally has two intentions. First, the doctor want to confirm his primary diagnosis according to the patient answers. Second, the doctor can rule out the possible disease based on the patient replies. The process of medical diagnosis is essential a step-by-step way of removing the candidate diseases. As shown in Figure 3, when to request a symptom, we want the distribution of disease to be deterministic, which means the distribution should have some peaks and valleys (shown in red distribution), rather than flat plains (shown in blue distribution).

Therefore, given the inference engine D_ϕ (i.e., we fixed the parameters ϕ during training of generator), the generated candidate symptom should enhance the mutual information reward R_M.

$$R_M = Q_{D_\phi}^{G_\theta}(s = Y'_{1:t-1}, a = y_t) = I(O_{t-1}; O_t|D_\phi) \tag{7}$$

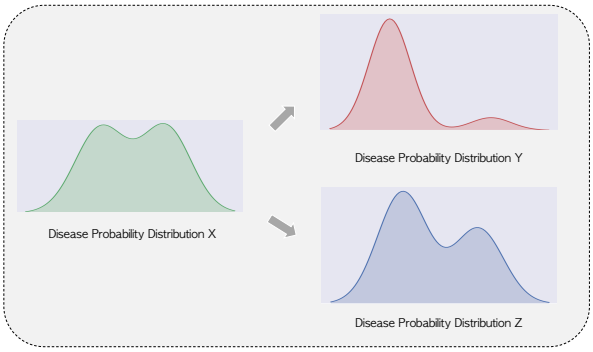

Figure 3: Intuitive Illustration of Mutual Information Reward for Medical Diagnosis. At the current state, the disease distribution is shown in green. When requesting a symptom, we prefer to get the red distribution rather than the blue one.

To make the generated symptom sequence to be both natural, authentic and capable of making differential disease, we update the policy parameters with the mixed R_M and R_D reward as R_F. Here, we have:

$$R_F = (1-\lambda)R_M + \lambda(R_D - \epsilon) \tag{8}$$

where λ controls the weight of the reward, ϵ controls the effect of discriminator. If a discriminator thinks the generated sequence is real, then the reward should be positive, ϵ is normally set to 0.5, as the discrimnator evaluates the sequence as real if the probablity is greater than 0.5.

When the evaluation discriminator D_ψ has been updated, we are ready to update the generator. The proposed policy based method relies upon optimizing a parametrized policy to directly maximize the long-term reward. Following (Sutton et al. 2000), the gradient of the objective function $J(\theta)$ w.r.t. the generators parameters θ can be derived as:

$$\nabla_\theta J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{y_t \sim G_\theta}[\nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot R_F^{(t)}] \tag{9}$$

Finally, we update the parameters of generator G_θ with as:

$$\theta := \theta + \alpha \nabla_\theta J(\theta) \tag{10}$$

Generator of Dialogue Agent A complete medical inquiry process normally contains the patient chief complaint (i.e. self-report) and present history. The doctor inquiry the patient based on his chief complaint. Then, the present history is accomplished in an interactive way, as the doctor asks the question and the patient confirms or denies. In our work, the sequence inquiry generator G_θ can be viewed as a doctor agent to inquiry the patient.

We use the recurrent neural network (RNNs) (Hochreiter and Schmidhuber 1997) as generator G_θ. The RNN maps the input inquiry sequence $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$ into a inquiry sequence of hidden state $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_T$ by using the update function σ recursively.

$$\boldsymbol{h}_t = \sigma(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \tag{11}$$

In general neural language model (Bengio et al. 2003), we normally apply a softmax output layer maps the hidden states into the output token distribution, where weight matrix W and vector b are parameters.

$$p(x_t|x_1, \cdots, x_{t-1}) = \text{softmax}(\boldsymbol{W}\boldsymbol{h_t} + \boldsymbol{b}) \tag{12}$$

Unfortunately, standard RNN suffers the problem of gradient vanishing or exploding, where gradients may grow or decay exponentially over long sequences. The advent of LSTM (Hochreiter and Schmidhuber 1997) is to deal with the weakness of standard RNN. We build the generator G_θ with an LSTM model, while the variant of RNNs, such as the gated recurrent unit GRU (Cho et al. 2014), attention-mechanism based RNN model (Bahdanau, Cho, and Bengio 2014), can also be used as a generator.

Discriminator and Classifier of Dialogue Agent We adopt a deep neural network (DNN) (Hinton and Salakhutdinov 2006) to train discriminator D_ψ and inference engine D_ϕ. It is worth noticing that other kinds of neural net architectures, such as the convolutional neural network (CNN) (Kim 2014) and recurrent neural network (RNN) can be used as a discriminator in our framework. The discriminator and the inference engine share the same architecture, except for the final output layer. The evaluation discriminator outputs a single scalar, which represents the probability that symptom sequence comes from the real data rather than the generator. Second, for the inference engine D_ϕ, we use the patient self-report, conversations and doctor's diagnosis from the training corpus to train a disease inference classifier. The output of $D_\phi(\cdot)$ represents the disease probability distribution.

The detailed training settings of our dialogue system is demonstrated in Experiment section.

Experiments

Datasets

To evaluate the performance of the proposed framework, we test our system on two public medical dialogue datasets, MuZhi Medical Dialogue dataset [1], and Dxy Medical Dialogue dataset [2].

MuZhi Medical Dialogue Dataset. The MuZhi dataset is constructed by (Liu et al. 2018), the data is collected from the pediatric department in a Chinese online healthcare website[3], which is a popular website for users to consult doctors online. Usually, the patient will provide a self-report to show his or her basic information. The doctor will then initiate a dialogue to gather more information and inference the possible disease based on self-report and the conversation. The doctor can obtain additional symptoms during the conversation, which are not mentioned in the self-report. For

[1] http://www.sdspeople.fudan.edu.cn/zywei/data/acl2018-mds.zip

[2] https://github.com/HCPLab-SYSU/Medical_DS

[3] https://muzhi.baidu.com

Method	Infantile diarrhea	Dyspepsia	Upper respiratory infection	Bronchitis	Overall
SVM-ex	0.89	0.28	0.44	0.71	0.59
SVM-ex&im	0.91	0.34	0.52	0.93	0.71
Basic DQN	-	-	-	-	0.65
DQN + relation branch	0.92	0.35	0.49	0.93	0.70
Our best result	0.88	0.62	0.72	0.67	**0.73**

Table 1: Performance comparisons on Muzhi dataset.

Method	Accuracy	Ave turns	Match rate
Basic DQN(Liu et al. 2018)	0.731	3.92	0.110
Sequicity (Lei et al. 2018)	0.285	3.40	0.246
KR-DS(Xu et al. 2019)	0.740	3.36	**0.267**
pretrained LSTM	0.643	3.47	0.132
Vanilla PG	0.731	3.11	0.212
PG + MI	0.749	2.76	0.159
PG + GAN	0.758	3.4	0.205
PG + MI-GAN	**0.769**	**2.68**	0.179

Table 2: Performance comparisons with the state-of-art methods on Dxy dataset.

each patient, there is a final diagnosis given by the doctor, which is defined as the label. The dataset defines the symptoms from self-reports as explicit symptoms and those from the dialogue between patients and doctors as implicit symptoms. This dataset contains 710 user goals and 66 symptoms, with four kinds of labeled diseases, including upper respiratory infection, children functional dyspepsia, infantile diarrhea, and childrens bronchitis. The dataset is labeled with the symptom phrases in both self-reports and conversational data by three annotators.

Dxy Medical Dialogue Dataset. The Dxy Dialogue dataset is collected from another Chinese online healthcare community[4] where users asking doctors professional medical advice. This dataset contains 527 conversational data in total. There are 423 conversational data to be selected as the training set, and 104 for testing. (Xu et al. 2019) annotate the dataset with five types of diseases, including allergic rhinitis, upper respiratory infection, pneumonia, children hand-foot-mouth disease, and pediatric diarrhea. The dataset extracts the symptoms that appear in self-reports and conversation. All the symptoms are normalized into 41 symptoms. The self-reports and raw conversations are labeled with four annotators who have a medical background. Similar to the MuZhi dataset, symptoms appearing in self-reports are defined as explicit symptoms while the others are implicit symptoms. The diseases of each medical diagnosis conversation are automatically extracted from the website.

Experiment Setup

Evaluation Metrics. The evaluation metrics contains diagnosis accuracy, average request turns and match rate, which is consistent with the previous work (Liu et al. 2018; Xu et al. 2019). Diagnosis accuracy and average turns are significant metrics. An excellent doctor can make a correct diagnosis in just a few rounds of consultation. The match rate, to some extent, is important, while the high match rate is not equivalent to high accuracy.

Training Setting. Our dialogue system has a generator G_θ, an evaluation discriminator D_ψ and an inference engine D_ϕ. All the parameters are initialized with normal distribution $\mathcal{N}(0, 0.01)$, and all neural network are train with the Adam optimizer (Kingma and Ba 2014). For the training of the generator G_θ, we pre-train the LSTM model in a language model method, which iteratively predicts the next symptom tokens. The batch size for pre-training LSTM is 128. The pre-training learning rate is set to 0.01. Then we use the pre-trained generator G_θ to make roll-outs to generate fake symptom sequences. In the architecture of GANs, the training set for the discriminator D_ψ is comprised of the generated examples with the label 0 and the instances from trainset with the label 1. For the training of GANs and policy network, we use the REINFORCE (Williams 1992) algorithm. We update the generator parameters with the discriminator's output and mutual information as a reward. The balance parameter λ is set to 0.5. The policy network G_θ learning rate is set to 0.0001 while learning rate of discriminator D_ψ is set to 0.01. Meanwhile, we train the disease inference engine D_ϕ by minimizing the cross-entropy loss on training set. The learning rate of D_ϕ is 0.001, the batch size is 64. The maximum turn T is set to 20. The threshold τ of inference engine is 0.8. Generally, we train 100 epochs, for every epoch the g-steps, d-steps, and k-steps in the Algorithm 1 is set to 2, 1, 25, respectively. The deep learning models are implemented in PaddlePaddle [5].

Experiment Results

Muzhi dataset. We first evaluate our proposed framework on Muzhi dataset. (Liu et al. 2018) use the SVM to train the "lower" and "upper" bound for this dataset. The basic DQN gets the accuracy between the SVM-ex and SVM-ex & im results. (Xu et al. 2019) use a knowledge-route branch and a relation branch method to improve diagnosis accuracy. Additional knowledge graph helps to increase the accuracy, while our method does not need extra knowledge graph. Therefore, we only compare our model with the DQN + relation branch. The experiment results are shown in Table 1. As shown in the table, our proposed method is superior to other frameworks. Note that the above DQN based method both have a bias on some disease, for they both get very high accuracy on infantile diarrhea and bronchitis (0.92 and 0.93 respectively), while getting low accuracy on dyspepsia and upper respiratory infection (0.35 and 0.49 respectively). The reason behind this phenomenon is that their models fail to ask the key symptoms which can distinguish the possible disease (i.e. the clinical manifestation of infantile diarrhea and bronchitis are similar). Our model uses the mutual information reward to update the symptom generator, the symptoms with differentiating capabilities are incentive to

[4]https://dxy.com/

[5]https://github.com/PaddlePaddle/Paddle

be asked. Therefore, our model can achieve relatively higher accuracy than others for all four kinds of disease.

Dxy dataset. We further evaluate our proposed framework on Dxy dataset. We compare our work with the baseline model and other state-of-art frameworks for medical dialogue system. To be consistent with their experiment results, we use the extracted and normalized symptom tokens in their works. As illustrated in Table 2, our proposed method outperforms Basic DQN (Liu et al. 2018), Sequicity (Lei et al. 2018) and Knowledge-Routed DQN (Xu et al. 2019) with higher diagnosis accuracy and shorter average turns.

For the basic DQN method (Liu et al. 2018), the agent only asks the symptoms with the largest positive reward, which sometimes leads to inquiry unreasonable and repeated symptoms. With the help of knowledge graph and pre-calculated symptom-disease relations, (Xu et al. 2019) alleviates the problem of basic DQN. While knowledge-routed based method can get a higher match rate, it cannot get high accuracy with shorter request turns. (Xu et al. 2019) suggest that higher matching rate is more reasonable, while using high-frequency symptoms also lead to higher matching rate. The essence is that the above dialogue generated policy is updated with predefined and fixed reward. The intuition of the reward is not clear, the training results can be varied in different reward value. (Lei et al. 2018) performs worse on this medical diagnosis task as they care more about the transition of dialogues while not considering the connections between symptom and disease.

Ablation Studies

To further demonstrate the effectiveness of our proposed framework, we conducted a series of ablation studies on Dxy medical dialogue dataset.

Pre-trained LSTM. Our symptom sequence generator G_θ is pre-trained with an LSTM model via using the maximum likelihood estimation (MLE) objective. For the sequence generation, a maximum likelihood trained model is normally regarded as the baseline.

Vanilla Policy Gradient. In our framework, the pre-trained LSTM generator is then updated through policy gradient with the reward (feedback) of the user simulator. Here we use the predefined fixed reward. The reward for correct and wrong diagnosis is R_+ and R_- respectively. The reward for correct match is R_m and for repeated request is R_r. We try different combinations of rewards, the best result is shown in Table 2.

Policy Gradient with MI. To let the generator inquiry distinguished symptoms, which is significant when making a differential diagnosis, we introduce mutual information (MI) as the reward to update the symptom generator. The mutual information reward is calculated through the Eq.(7).

Policy Gradient with GAN. To let the generated symptom sequence appear to be natural, and not request strange symptoms, our work uses the discriminator's output probability as a reward to update the inquired policy, the reward is computed by the Eq.(2)

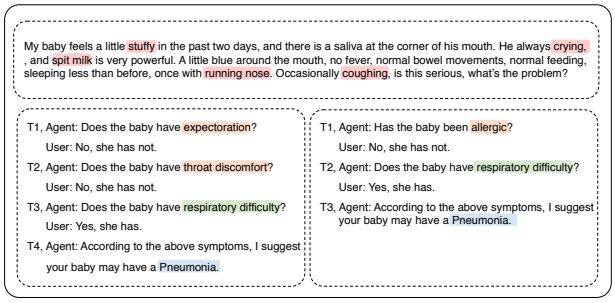

Figure 4: Visualization of the results on our proposed framework for Automatic Diagnosis. Red color indicates the explicit symptoms of the patient. Green color indicates the matched symptoms while orange color indicates the symptoms failed to match. Blue color indicates the predicted diagnosis. The top table is the user self-report from the Dxy dataset. The left table is the results from PG+GAN, while the right one is from PG+MI-GAN.

Policy Gradient with MI and GAN. Finally, we put all the components together. On the one hand, we want the inquired symptom with differentiating capabilities, and on the other hand, we want the generated sequence to be natural without strange or unreasonable symptom. Therefore, we update the generator with the mixed reward from discriminator reward and mutual information reward by Eq.(8).

The results are shown in Table 2. From the table, we can see that the performance of MLE based pre-trained LSTM is the worst. Because there is no direct instruction to tell the model what is the correct diagnosis. There is no interaction between the patient and the model. The results of the policy gradient method are similar to DQN, for the reason, it only has fixed the reward. The performance of PG+MI is superior to the vanilla PG, because the reward given by maximizing mutual information from disease distribution, can encourage the model to request the critical symptoms.

With the help of GANs, we can utilize the output of discriminator to evaluate the current generated sequence is real or fake. It can learn the latent inquiry patterns, the reward encourages the model to request symptoms in a way similar to a doctor. The final result shows that the GAN+MI based policy gradient framework is superior to all others, which get higher diagnosis accuracy with shorter average turns. As shown in Fig 4, both PG+GAN and PG+MI-GAN methods can make a correct diagnosis, and request the key symptom (respiratory difficulty), while the PG+MI-GAN method takes shorten turns.

Conclusions

In this work, we propose a Generative Adversarial regularized Mutual information Policy gradient framework (GAMP) for automatic diagnosis which aims to make a better medical dialogue system with higher diagnosis accuracy and less interactive turns with the user. First, we propose a new technique, called generative adversarial regularized policy gradient, to optimize the diagnosis system, which

tries to avoid inquiring unreasonable symptoms deviate from the doctor's common diagnosis paradigm. Second, we devise a mechanism to add mutual information as a part of the reward function. Experiment evaluations on two public datasets have confirmed the validity of our proposed method. It not only can improve the accuracy of diagnosis but also can use less inquires to make a diagnosis decision.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. Journal of machine learning research 3(Feb):1137–1155.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2172–2180.

Chen, J.; Zhou, J.; Shi, Z.; Fan, B.; and Luo, C. 2019. Knowledge abstraction matching for medical question answering. In BIBM.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Eric, M., and Manning, C. D. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. arXiv preprint arXiv:1701.04024.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In NIPS, 2672–2680.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. science 313(5786):504–507.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural computation 9(8):1735–1780.

Kao, H.-C.; Tang, K.-F.; and Chang, E. Y. 2018. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In AAAI.

Kim, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In ACL, 1437–1447.

Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547.

Liu, B.; Tur, G.; Hakkani-Tur, D.; Shah, P.; and Heck, L. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. arXiv preprint arXiv:1711.10712.

Liu, Q.; Wei, Z.; Peng, B.; Tou, H.; Chen, T.; Huang, X.; Wong, K.-F.; and Dai, X. 2018. Task-oriented dialogue system for automatic diagnosis. In ACL, volume 2, 201–207.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In ICML, 2642–2651. JMLR. org.

Peng, Y.-S.; Tang, K.-F.; Lin, H.-T.; and Chang, E. 2018. Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis. In NIPS, 7322–7331.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. nature 529(7587):484.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. Nature 550(7676):354.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In NIPS, 3104–3112.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In NIPS, 1057–1063.

Tang, K.-F.; Kao, H.-C.; Chou, C.-N.; and Chang, E. Y. 2016. Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In NIPS Workshop on Deep Reinforcement Learning.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8(3-4):229–256.

Xu, L.; Zhou, Q.; Gong, K.; Liang, X.; Tang, J.; and Lin, L. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In AAAI.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In AAAI.

Yu, C.; Liu, J.; and Nemati, S. 2019. Reinforcement learning in healthcare: A survey. arXiv preprint arXiv:1908.08796.