Task-oriented Dialogue System for Automatic Disease Diagnosis via Hierarchical Reinforcement Learning

Kangenbei Liao[1], **Qianlong Liu**[2], **Zhongyu Wei**[1*], **Baolin Peng**[3], **Qin Chen**[1], **Weijian Sun**[4], **Xuanjing Huang**[5]

[1]School of Data Science, Fudan University, China
[2]Alibaba Group, China
[3]Microsoft Research, USA
[4]Huawei Technologies Co., Ltd., China
[5]School of Computer Science, Fudan University, China
{18210980053, zywei,qin_chen}@fudan.edu.cn, qianlong.lql@alibaba-inc.com,
bapeng@microsoft.com, sunweijian@huawei.com, xjhuang@fudan.edu.cn

Abstract

In this paper, we focus on automatic disease diagnosis with reinforcement learning (RL) methods in task-oriented dialogues setting. Different from conventional RL tasks, the action space for disease diagnosis (i.e., symptoms) is inevitably large, especially when the number of diseases increases. However, existing approaches to this problem employ a flat RL policy, which typically works well in simple tasks but has significant challenges in complex scenarios like disease diagnosis. Towards this end, we propose to integrate a hierarchical policy of two levels into the dialogue policy learning. The high level policy consists of a model named master that is responsible for triggering a model in low level, the low level policy consists of several symptom checkers and a disease classifier. Experimental results on both self-constructed real-world and synthetic datasets demonstrate that our hierarchical framework achieves higher accuracy in disease diagnosis compared with existing systems. Besides, the datasets [1] and codes[2] are all available now.

1 Introduction

With the development of electronic health records (EHRs) systems, researchers explore different machine learning approaches for automatic diagnosis [1]. Although impressive results have been reported for the identification of various diseases [2; 3], they rely on well established EHRs which are labor-intensive to build. Moreover, supervised model trained for one disease is difficult to be transferred to another, therefore, EHRs are needed for every single disease.

In order to relieve the pressure for constructing EHRs, researchers [4; 5] introduce task-oriented dialogue system to request symptoms automatically from patients for disease diagnosis. They formulate the task as Markov Decision Processes (MDPs) and employ reinforcement learning (RL) based methods for the policy learning of the system. Existing framework utlizes a setting of flat policy that treats diseases and all related symptoms equally. Although RL-based approaches have shown positive results for symptom acquisition, when it comes to hundreds of diseases in real environment, the setting of flat policy is quite impractical.

In general, a particular disease is related to a certain group of symptoms. That's to say, a person who suffers a disease will often carries some corresponding symptoms at the same time. As shown in figure 1, we present the correlation between diseases and symptoms. x-axis represents symptoms and y-axis is the proportion of diseases related. We can easily identify some patterns. In other word, each disease has a group of corresponding symptoms and the overlap among different groups of symptoms are limited. This motivates us to classify diseases into different groups following the setting of departments in the hospital and design a hierarchical structure for symptom acquisition and disease diagnosis.

Recently, Hierarchical Reinforcement Learning (HRL) [6; 7], in which multiple layers of policies are trained to perform decision making, has been successfully applied to different scenarios, including course recommendation [8], visual dialogue [9], relation extraction [10], etc. The natural hierarchy of target tasks are modeled either manually or automatically. Inspired by these research, we explore to utlize the clustering information of diseases via HRL to deal with the issue of large action space.

In this paper, we classify diseases into several groups and build a dialogue system with a hierarchy of two levels for automatic disease diagnosis using HRL methods. The high level policy consists of a model named master and the low level policy consists of several workers and a disease classifier. The master is responsible for triggering a model in the low level. Each worker is responsible for inquiring symptoms related to a certain group of disease while disease classifier is responsible for making the final diagnosis based on information collected by workers. The proposed framework imitates

*Corresponding Author

[1]http://www.sdspeople.fudan.edu.cn/zywei/data/Fudan-Medical-Dialogue2.0

[2]https://github.com/nnbay/MeicalChatbot-HRL

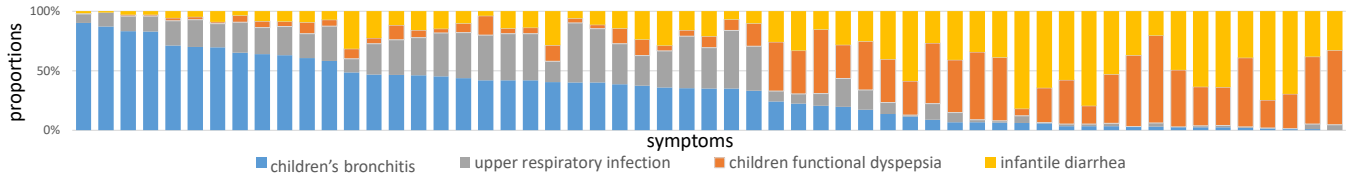

Figure 1: The disease distribution over symptoms in the real-world dataset (see section 3.1). x-axis stands for symptoms and y-axis is the proportion. Each bar describes the disease distribution given a symptom.

a group of doctors from different departments to diagnose a patient together. Among them, each worker acts like a doctor from a specific department, while the master acts like a committee that appoints doctors to interact with this patient. When information collected from workers are sufficient, the master would activate a separate disease classifier to make the diagnosis. Models in the two levels are trained jointly for better disease diagnosis. We build a large real-world dataset and a synthetic dataset for the evaluation of our model. Experimental results demonstrate that the performance of our hierarchical framework outperforms other state-of-the-art approaches on both datasets.

2 Hierarchical Reinforcement Learning Framework for Disease Diagnosis

In this section, we introduce our hierarchical reinforcement learning framework for disease diagnosis. We start with the flat policy setting and then introduce our hierarchical policy with two levels. We further improve the performance of our model via reward shaping techniques.

2.1 Reinforcement Learning Formulation for Disease Diagnosis

As for RL-based models for automatic diagnosis, the action space of agent $\mathcal{A} = D \cup S$, where D is the set of all diseases and S is the set of all symptoms that associated with theses diseases. Given the state $s_t \in \mathcal{S}$ at turn t, the agent takes an action according to it's policy $a_t \sim \pi(a|s_t)$ and receives an immediate reward $r_t = R(s_t, a_t)$ from the environment (patient/user). If $a_t \in S$, the agent chooses a symptom to inquire the patient/user. Then the user responds to the agent with *true/false/unknown* and the corresponding symptom will be represents via a 3-dim one-hot vector $b \in \mathbb{R}^3$ accordingly. If $a_t \in D$, the agent informs the user with the corresponding disease as the diagnosis result and the dialogue session will be terminated as success/fail in terms of the correctness of diagnosis. The state $s_t = [b_1^\top, b_2^\top, \cdots, b_{|S|}^\top]^\top$, i.e., the concatenation of one-hot encoded statuses of each symptom, and not-requested symptoms until turn t are all encoded as $b = [0, 0, 0]$.

The goal for the agent is to find an optimal policy so that it can maximizes the expected cumulative future discounted rewards $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r'_t$, where $\gamma \in [0, 1]$ is the discounted factor and T is maximal turn of current dialogue session. The Q-value function

$$Q^\pi(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$$

is the expected return of taking action a in state s following a policy π.

The optimal Q-value function is the maximum Q-value among all possible policies: $Q^*(s, a) = \max_\pi Q^\pi(s, a)$. It follows the Bellman equation:

$$Q^*(s, a) = \mathbb{E}_{s'}[r + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a') | s, a]$$

A policy π is optimal if and only if for every state and action, $Q^\pi(s, a) = Q^*(s, a)$. Then the policy can be reduced deterministically by $\pi(a|s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a)$.

2.2 Hierarchical Policy of Two Levels

In order to reduce the problem of large action space, we extend the above RL formulation to a hierarchical structure with two-layer policies for automatic diagnosis. Following the *options* framework [7], our framework is designed as in Figure 2. There are four components in five framework: master, workers, disease classifier, internal critic and user simulator.

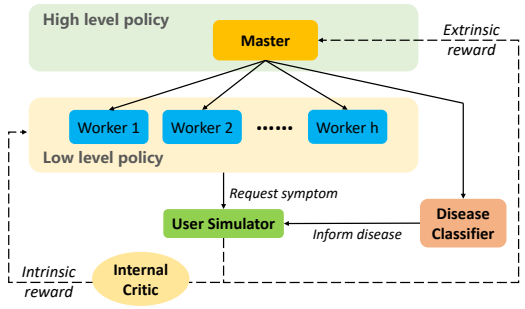

Figure 2: The framework of our hierarchical reinforcement learning model with two-layer policies.

Specifically, we divide all the diseases in D into h subsets $D_1, D_2, \ldots, D_h$, where $D_1 \cup D_2 \cup \cdots \cup D_h = D$ and $D_i \cap D_j = \varnothing$ for any $i \neq j$ and $i, j = 1, 2, \cdots, h$. Each D_i is associated with a set of symptoms $S_i \subseteq S$, whose symptoms are related to diseases in D_i. While worker w^i is responsible for collecting information from user about symptoms of S_i.

At turn t, the master decides whether to collect symptom information from user (picking one worker to interact with user for several turns) or inform the user with diagnosis result (picking disease classifier to output the predicted disease). An illustration of the diagnosis process with interactions between models in two levels are presented in Figure 3. As for internal critic, it is responsible for both returning intrinsic reward to the worker and telling whether the subtask of the invoked worker is finished. In addition, the user simulator is applied to interact with our model and return extrinsic reward.

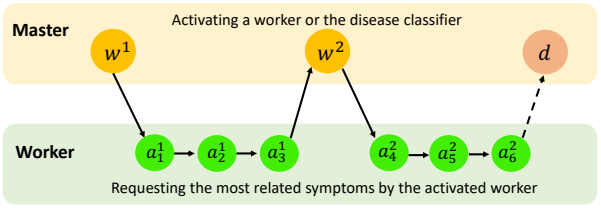

Figure 3: Illustration of the diagnosis process of our model with interactions between models in two levels. w^i is the action invoking worker w^i and d is the action invoking disease classifier.

Master

The action space of master $\mathcal{A}^m = \{w^i | i = 1, 2, \cdots, h\} \cup \{d\}$. The action w^i indicates activating worker w^i while d is a primitive action which means activating the disease classifier. At each turn t, the master takes the dialogue state $s_t \in \mathcal{S}$ as input and takes an action $a_t^m \in \mathcal{A}^m$ according to it's policy $\pi^m(a_t^m|s_t)$. An extrinsic reward r_t^e will be returned to master from the environment.

The decision process of master is not a standard MDP. Once the master activates a worker, this worker will interact with user for N turns until the subtask is terminated. Only after that master can take a new action and observe a new dialogue state. As pointed out by [7], the learning problem of master can be formulated as a Semi-Markov Decision Process (SMDP), where the extrinsic rewards returned during the interaction between user and the chosen worker can be accumulated as the immediate rewards for the master [11]. That is to say, after taking an action a_t^m, the reward r_t^m for master can be defined as:

$$r_t^m = \begin{cases} \sum_{t'=1}^{N} \gamma^{t'} r_{t+t'}^e, & if \quad a_t^m = w^i \\ r_t^e, & if \quad a_t^m = d \end{cases}$$

where $i = 1, ..., h$, r_t^e is the extrinsic reward returned by the environment at turn t, γ is the discounted factor of the master and N is the number of primitive actions of worker. We can write the Bellman equation for master as follows:

$$Q_m(s, a^m) = r^m + \mathbb{E}_{\{s', a^{m\prime}\}} \left[\gamma^N Q_m(s', a^{m\prime}) | s, a^m, \pi^m\right]$$

Where s' is the observed dialogue state of master after it takes an action a^m, $a^{m\prime}$ is the next action when the state is s'.

The objective of master is to maximize the extrinsic reward through SMDP, thus we can write the master's loss function as follows:

$$\mathcal{L}(\theta_m) = \mathbb{E}_{s, a^m, r^m, s' \sim \mathcal{B}^m}[(y - Q_m(s, a^m; \theta_m))^2]$$

where $y = r^m + \gamma^N \max_{a^{m\prime}} Q_m(s', a^{m\prime}; \theta_m^-)$, θ_m is the network parameter at current iteration, θ_m^- is the network parameter of previous iteration and $\mathcal{B}^m$ is the fixed-length buffer of samples for master.

Worker

The worker w^i corresponds to the set of diseases D_i and the set of symptoms S_i. The action space of worker w^i is:$\mathcal{A}_i^w = \{\texttt{request}(symptom) | symptom \in S_i\}$. At turn t, if worker w^i is invoked, the current state of master s_t will be passed to worker w^i, then worker w^i will extract s_t^i from s_t and take s_t^i as input and generate an action $a_t^i \in \mathcal{A}_i^w$. The state extraction function is as follows:

$$s_t^i = ExtractState(s_t, w^i) = [b_{(1)}^{i\top}, b_{(2)}^{i\top}, \ldots, b_{(k_i)}^{i\top}]^\top$$

where $b_{(j)}^i$ is the representing vector of $symptom_{(j)} \in S_i$.

After taking action $a_t^i \in \mathcal{A}_i^w$, the dialogue is updated into s_{t+1} and worker w^i will receive an intrinsic reward $r_t^{\mathtt{i}}$ from the module of internal critic. So the objective of worker is to maximize the expected cumulative discounted intrinsic rewards. The loss function of worker w^i can be written as:

$$\mathcal{L}(\theta_w^i) = \mathbb{E}_{s^i, a^i, r^{\mathtt{i}}, s^{i\prime} \sim \mathcal{B}_i^w}[(y_i - Q_w^i(s^i, a^i; \theta_w^i))^2]$$

where $y_i = r^{\mathtt{i}} + \gamma_w \max_{a^{i\prime}} Q_w^i(s^{i\prime}, a^{i\prime}; \theta_w^{i-})$, γ_w is the discounted factor of all the workers, θ_w^i is the network parameter at current iteration, θ_w^{i-} is the network parameter of previous iteration and $\mathcal{B}_i^w$ is the fixed-length buffer of samples for worker w^i.

Disease Classifier

Once the disease classifier is activated by master, it will take the master state s_t as input and output a vector $\mathbf{p} \in \mathbb{R}^{|D|}$, which represents the probability distribution over all diseases.The disease with highest probability will be returned to the user as the diagnosis result. Two layers of Multi-Layered Perceptron (MLP) is utilized here for the disease diagnosis.

Internal Critic

The internal critic is responsible for generating intrinsic reward $r_t^{\mathtt{i}}$ to worker w^i after an action a_t^i is taken at turn t. $r_t^{\mathtt{i}}$ equals +1, if the worker requests a symptom that the user suffers. If there are repeated actions generated by worker w^i or the number of subtask turns reaches T^{sub}, $r_t^{\mathtt{i}}$ would be -1. Otherwise, $r_t^{\mathtt{i}}$ would be 0.

The internal critic is also responsible for judging the termination condition for the worker. In our task, a worker is terminated as failed when there is repeated action generated or the number of subtask turns reaches T^{sub}. While a worker is terminated as successful when the user responds $true$ to the symptom requested by the agent. Which means the current worker finishes the subtask by collecting enough symptom information.

User Simulator

Following [4] and [5], we use a user simulator to interact with the agent. At the beginning of each dialogue session, the user simulator samples a user goal from the training set randomly. All the explicit symptoms of the sampled user goal are used for initializing a dialogue session. During the course of dialogue, the simulator interacts with the agent based on the user goal following some rules. One dialogue session will be terminated as successful if the agent make the correct diagnosis. It will be terminated as failed if the informed disease is incorrect or the dialogue turn reaches the maximal turn T. In order to improve the efficiency of the interaction, the dialogue would be terminated when repeated action is taken by the system.

2.3 Reward Shaping

In reality, the number of symptoms a patient suffers from is much less than the size of symptom set S and this results in a sparse feature space. In other words, it is hard for the agent to locate the symptoms that the user truly suffers from. In order to encourage master to choose a worker that can discover more positive symptoms, we follow [12] and use the reward shaping method to add auxiliary reward to the original extrinsic reward while keeping the optimal reinforcement learning policy unchanged.

The auxiliary reward function from state s_t to s_{t+1} is defined as $f(s_t, s_{t+1}) = \gamma\phi(s_{t+1}) - \phi(s_t)$, $\phi(s)$ is the potential function and can be defined as

$$\phi(s) = \begin{cases} \lambda \times \big|\{j : b_j = [1,0,0]\}\big|, & \text{if } s \in \mathcal{S}/\mathcal{S}_\perp \\ 0, & \text{otherwise} \end{cases}$$

Where $\phi(s)$ counts the number of *true* symptoms for a given state s, $\lambda > 0$ is a hyper-parameter which controls the magnitude of reward shaping and $\mathcal{S}_\perp$ is the terminal state set. Thus, the reward function for master will be changed into $R_t^\phi = r_t + f(s_t, s_{t+1})$.

2.4 Training

Both the master policy π^m and each worker's policy π_i^w are parameterized via Deep Q-Network [13; 14]. In DQN, the action is often selected following an ϵ-greedy policy. In our hierarchical framework, both the master and the workers behave following their own ϵ-greedy policy for training and greedy policy for testing. During the training process, we store $(s_t, a_t^m, r_t^m, s_{t+N})$ in $\mathcal{B}^m$ and $(s_t^i, a_t^w, r_t^i, s_{t+1}^i)$ in $\mathcal{B}_i^w$. At each training step, we perform experience replay to update the current networks for both master and workers in $\mathcal{B}^m$ and $\mathcal{B}_i^w$ respectively, while the target networks are fixed during experience replay. The target network will be updated (replaced by the current network) only when one experience replay is over. At each step, the current network will be evaluated on the training set, the experience buffer will be flushed only if the current network performs better than any previous versions in success rate. Therefore, the samples generated in the previous iterations will be removed from the experience buffer and it will speed up the training process. As for disease classifier, it will be updated with terminal states and corresponding disease labels after every 10 epochs' training of master.

3 Dataset

We construct two datasets for the evaluation of our model. One is an extended version of an existing real-world dataset. Another is a synthetic dataset.

3.1 Real-world Dataset

There is an existing dataset collected from real world for the evaluation of task-oriented DS for diagnosis [4]. We extend the original dataset following their labeling strategy. The newly constructed real-world dataset (RD) contains 1,490 user goals that belong to 4 diseases, namely, upper respiratory infection (URI), children functional dyspepsia (CFD), infantile diarrhea (ID) and children's bronchitis (CB). The raw data is collected from the pediatric department on a Chinese Online Healthcare Community[3].

disease	♯ of user goal	ave. ♯ of ex. sym.	ave. ♯ of im. sym.	♯ of sym.
ID	450	2.22	4.68	83
CFD	350	1.73	5.05	81
URI	240	2.79	5.00	81
CB	450	3.01	5.35	84
Total	1490	2.44	3.68	90

Table 1: Overview of Real-world dataset (RD). ♯ of user goal is the number of dialogue sessions of each disease; ave. ♯ of ex. sym. and ave. ♯ of im. sym. are the average number of explicit and implicit symptoms of user goals; ♯ of sym. is the total number of symptoms that related to the disease.

Each user record consists of the self-report provided by the user and conversation text between the patient and a doctor. We hire experts with medical background to identify symptom expressions and label them with three tags ("True", "False" or "UNK") to indicate whether the user suffers this symptom. After then, experts manually link each symptom expression to a concept on SNOMED CT [4]. Note that, both self-reports and the conversations are labeled. Symptoms extracted from self-report are treated as explicit symptoms and the ones extracted from conversation are implicit symptoms. Statistics of RD dataset can be seen in Table 1.

3.2 Synthetic Dataset

In addition to the real-world dataset, we build a synthetic dataset (SD) following [15]. It is constructed based on symptom-disease database called SymCat[5]. There are 801 diseases in the database and we classify them into 21 departments (groups) according to International Classification of Diseases (ICD-10-CM)[6]. We choose 9 representative departments from the database, each department contains top 10 diseases according to the occurrence rate in the Centers for Disease Control and Pre-vention (CDC) database.

In CDC databse, each disease is linked with a set of symptoms, where each symptom has a probability indicating how likely the symptom is identified for the disease. Based on the probability distribution, we generate record one by one for each target disease. Given a disease and its related symptoms, the generation of a user goal follows two steps. First, for each related symptom, we sample the label for the symptom (true or false). Second, a symptom is chosen randomly from the set of all true symptoms to be the explicit one (same as symptoms extracted from self-report in RD) and rest of true symptoms are treated as implicit ones. A generated record for SD dataset can be seen in Table 6. The description of SD dataset is shown in Table 2. The synthetic dataset we constructed contains 30,000 user goals, of which 80% for training and 20% for testing.

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

[4] https://www.snomed.org/snomed-ct

[5] www.symcat.com

[6] https://www.cdc.gov/nchs/icd/

group id	♯ of user goal	♯ of diseases	ave. ♯ of im. sym.	♯ of sym.
1	3,371	10	3.23	65
4	3,348	10	1.71	89
5	3,355	10	2.67	68
6	3,380	10	2.83	58
7	3,286	10	2.78	46
12	3,303	10	2.04	51
13	3,249	10	2.48	62
14	3,274	10	1.58	69
19	3,389	10	2.91	73
Total	30,000	90	2.60	266

Table 2: Overview of the Synthetic Dataset (SD). Each user goal contains only 1 explicit symptom. The group id is correspond to the chapter in ICD-10-CM; ♯ of diseases is the number of diseases included in this group.

4 Experiments and Results

4.1 Implementation Details

The ϵ for master and all the workers are all set to 0.1. For the master, the maximal dialogue turn T is set to 20, it will receive a extrinsic reward of +1 if the master inform the right disease. Otherwise, it will receive a extrinsic reward of -1 if the dialogue turn reaches the maximal turn or a wrong disease is informed. At non-terminal turns, the extrinsic reward is 0. Moreover, the hyperparameter λ in reward shaping is set to +1, the sum of the extrinsic rewards (after reward shaping) over one subtask taken by a worker will be the reward for master. The maximal dialogue turn T^{sub} is set to 5 for each worker. For master and all the workers, the neural network of DQN is a three-layer network with two dropout layers and the size of hidden layer is 512. The discounted rate γ is set to 0.95 for both master and workers, learning rate is set to 0.0005. All parameters are set empirically and settings for the two datasets are the same. In addition, all the workers are trained every 10 epochs during the training process of master. For the disease classifier, the neural network is a two-layer network with a dropout layer and the size of hidden layer is 512, learning rate is set to 0.0005. It's trained every 10 epochs during the training process of master.

4.2 Models for Comparison

We compare our model with some state-of-the-art reinforcement learning models for disease diagnosis.

- *Flat-DQN*: This is the agent of [4], which has one layer policy and an action space including both symptoms and diseases.
- *HRL-pretrained*: This is a hierarchical model from [15]. The setting is similar to ours, however, the low level policy is pre-trained first and then the high level policy is trained. Besides, there is no disease classifier for disease diagnosis specially and the diagnosis is made by workers.

Note that, for RL setting, the user goals are initialized with explicit symptoms, while implicit symptoms can only obtained via conversations.

In addition, we implement two models following supervised learning setting that treats the automatic diagnosis as a multi-class classification problem. We report results of these two models for reference.

- *SVM-ex&im*: This model takes symptoms representation (concatenation of b_j, where $j = 1, 2, \cdots, |S|$) as input and predicts the target disease. Both explicit and implicit symptoms are used as input and SVM is used for classification. Because it obtains all implicit symptoms from the user, this model can be treated as the up-bound of RL-based models.
- *SVM-ex*: This model takes only explicit symptoms as input and use SVM for classification. It can be treated as the baseline of RL-based models.

4.3 Overall Performance

For both RD dataset and SD dataset, 80% of samples are used as training and 20% are used as testing. We use three metrics for the evaluation of the dialogue system following [4] and [5], namely, success rate, average reward and the average number of turns per dialogue session. Note that the result of *HRL-pretrained* on RD dataset is missing because there is only one disease in each group and it is non-trivial to implement the multi-classification model for disease diagnosis.

Table 3 and Table 4 show the overall performance of different models on RD dataset and SD dataset respectively. We have following findings:

- SVM-ex&im outperforms SVM-ex greatly on both two datasets, which indicates that implicit symptoms can improve the diagnosis accuracy significantly. Moreover, due to the lower overlap of symptoms between different diseases, the gap between SVM-ex&im and SVM-ex on SD dataset is much larger than the gap on RD dataset.
- Due to the additional requested implicit symptoms, the Flat-DQN model, HRL-trained model and our HRL model reach a better diagnosis accuracy than SVM-ex on both datasets, which proves the efficiency by introducing reinforcement learning based models.
- Compared to the baseline models, our HRL model takes more turns to have interactions with the user so that it can collect more information of their implicit symptoms. With more inforamtion about implicit symptoms, our HRL model significantly outperforms the other baselines in the diagnosis success rate.

Model	Success	Reward	Turn
SVM-ex	0.663±.003	\	\
Flat-DQN	0.681±0.018	0.509±0.029	2.534±0.171
HRL-pretrained	\	\	\
ours	0.695±0.018	0.521±0.022	4.187±0.187
SVM-ex&im	0.761±.006	\	\

Table 3: Overall performance on RD dataset. The experiment is carried by 5 times and the final result is composed of the average and the standard error of 5 experiments.

4.4 Further Analysis

In order to evaluate the performance of different workers and disease classifier, we perform some additional experiments based on our HRL model.

Model	Success	Reward	Turn
SVM-ex	0.321±.008	\	\
Flat-DQN	0.343±0.006	0.327±0.003	2.455±0.065
HRL-pretrained	0.452±0.013	0.439±0.023	6.838±0.358
ours	0.504±0.018	0.473±0.056	12.959±0.704
SVM-ex&im	0.732±.014	\	\

Table 4: Overall performance on SD dataset. The experiment is carried by 5 times and the final result is the average and the standard error of 5 experiments.

Performance of Disease Classifier

In order to have deeper analysis of the user goals which have been informed the wrong disease by the agent, we collect all the wrong informed user goals and present the error matrix in Fig 4. It shows the disease prediction result for all the 9 groups. We can see the color of the diagonal square is darker than the others, which means most of wrong informed user goals are informed the disease in the same group. This is reasonable because diseases in the same groups usually share similar symptoms and are therefore more difficult to be distinguished.

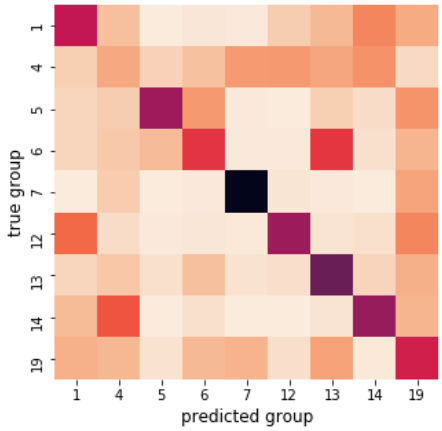

Figure 4: The error analysis for the disease classifier in different groups on our HRL model, the square with true group i and predicted group j means a disease in group i is misclassified into group j by the disease classifier, the darker the color, the higher the value.

Performance of Different Workers

We evaluate the performance of workers in terms of success rate, average intrinsic rewards and match rate. Match rate means the proportion of actions which have requested about the implicit symptoms that the user has. The results can be seen in Table 5. We can see there is positive correlation between the average intrinsic reward and the match rate, which means the more implicit symptoms a worker has requested from the user, the better of its performance.

Dialogue Case Study

In order to compare the performance between the reinforcement learning based models, we choose a user goal in the test set of SD dataset and output the dialogue content. The user goal we chose is shown in Table 6:

group id	success rate	ave intrinsic reward	match rate	activation times
1	48.6%	0.031	16.74%	0.615
4	54.6%	-0.150	5.02%	0.375
5	38.8%	-0.013	7.96%	3.252
6	48.0%	-0.036	9.58%	0.942
7	48.3%	0.057	18.57%	1.280
12	43.0%	0.021	11.26%	0.666
13	52.4%	-0.138	7.18%	0.823
14	72.2%	-0.111	3.77%	0.614
19	47.4%	0.031	22.72%	1.124
Average	50.3%	-0.041	10.49%	1.077

Table 5: The performance of different workers in our HRL model.

disease tag: Cerebral edema	**group id: 6**
explicit symptom	**implicit symptom**
headache: True	focal weakness: True diminished vision: True vomiting: True loss of sensation: True

Table 6: The selected user goal in SD dataset. This user goal contains 1 explicit symptom and 4 implicit symptoms in total. In addition, the corresponding disease is in group 6.

Firstly, we output the dialogue in the user goal according to the trained Flat-DQN model.

dialogue turn	agent action	user action
1	Inform the disease of Concussion.	Over

Table 7: The output dialogue of Flat-DQN model. It only lasts one turn and comes to an end.

From Table 7, we can see that the agent inform the disease to the user at the first turn without requesting any symptoms, which results in the wrong prediction before collecting enough information.

Secondly, we output the dialogue in the user goal according to the HRL-trained model.

From Table 8, we can see that HRL-trained model has more turns compared to Flat-DQN model. Moreover, the agent requests 9 symptoms from the user in total, while only one symptom the user truly has. From the worker id column, the worker id selected at each turn changes a lot while the worker 6 appears most, which is also the group id in the user goal. At the last turn, even though the agent selects the right worker, it doesn't inform the right disease to the agent, which results in a fail dialogue.

Finally, we output the dialogue in the user goal according to our trained HRL model.

From Table 9, we can see that compared with Flat-DQN model, our model has more turns to interact with the user. When compared with HRL-trained model, our model request more symptoms that the user truly has, which results in a more accurate diagnosis. From the worker id column, the master agent activates the worker 5 and worker 6 separately. Even though at first the wrong worker is activated, it still col-

dialogue turn	worker id	agent action	user action
1	1	Do you have skin rash?	No
2	6	Do you have neck pain?	No
3	5	Do you have anxiety and nervousness?	No
4	5	Do you have depressive or psychotic symptoms ?	No
5	6	Do you have Abnormal involuntary movements?	No
6	7	Do you have Eye redness?	No
7	13	Do you have Sharp chest pain?	No
8	7	Do you have diminished hearing?	No
9	6	Do you have loss of sensation?	Yes
10	6	Inform the disease of encephalitis.	Over

Table 8: The output dialogue of HRL-trained model. It requests 9 symptoms from the user and finally inform the disease.

dialogue turn	worker id	agent action	user action
1	5	Do you have neck pain?	No
2	5	Do you have ache all over?	No
3	5	Do you have vomiting?	Yes
4	6	Do you have depressive or psychotic symptoms ?	No
5	6	Do you have nausea?	No
6	6	Do you have dizziness?	No
7	6	Do you have anxiety and nervousness?	No
8	6	Do you have loss of sensation?	Yes
9	/	Inform the disease of Cerebral edema.	Over

Table 9: The output dialogue of our HRL model. It activates two different workers and requests 8 symptoms from the user. Finally it informs the right disease to the user.

lects some information and then activate the right worker at turn 4. At turn 9, the disease classifier is activated and output the right disease to the user.

5 Related Work

This paper is related to two major research fields, namely hierarchical reinforcement learning and RL-based methods for disease diagnosis.

Hierarchical reinforcement learning HRL has a hierarchical policy and has been proposed to solve the problems with large action space. One classic framework of HRL is *options* framework [7], which involves abstractions over action space. At each step, the agent chooses either a one-step "primitive" action or a "multi-step" action (option). [13] proposed a hierarchical-DQN, which integrates deep Q-learning into HRL, to learn policies for different levels simultaneously. HRL has been successfully applied to different tasks and reached promising results [8; 16; 9; 17; 10; 18]. Most existing works decompose the corresponding task into two steps manually, where the first step is finished by high level policy while the second step is finished by the low level policy. The task-specific design limits the generalization of HRL to complex tasks. [19] proposed a general framework that first learns useful skills (high level policies) in an environment and then leverages the acquired skills for learning faster in downstream tasks. What's more, some methods that generate or discover goals automatically when training the policies of two levels have been proposed [20; 21; 22].

RL-based methods for disease diagnosis There are some previous works that applies the flat RL-based method in the medical dialogue system [4; 5; 23; 15; 12] and generates positive results. In the work of [4] and [5], both symptoms and diseases are treated as actions and a flat policy is used for choosing actions. Considering the grouping between different diseases, [23] divide diseases into different groups based on anatomy and trained a policy for each group of diseases. Moreover, a rule-based method is used to choose a policy when interacting with patients. Based on the work of [23], [15] train a policy while fixing the policies of different groups to replace the rule-based method. Due to the separate training of high level and low level policy, the whole system may reach a sub-optimal solution. In addition, the action of informing disease to the user is taken by the low level policy rather than the high level policy, which can only have the diagnosis based on some limited information.

6 Conclusions and Future Work

In this work, we formulate disease diagnosis as a hierarchical policy learning problem, where symptom acquisition and disease diagnosis are assigned to different kinds of workers in the lower level of the hierarchy. A master model is designed in the higher level that is responsible for triggering models in low level. We extend a real-world dataset and build a synthetic dataset to evaluate our hierarchical model. To the best of our knowledge, this is the first time both kinds of datasets are used for model evaluation and we make both datasets public. The experimental results on both datasets demonstrate that our hierarchical model outperforms model with single layer of policy and is also better than another HRL model.

In the future, we would like to continue our research in three directions. First, we would like to make contribution on real-world dataset construction by introducing more diseases. Second, we will look into the module of natural language understanding and generation within the dialogue system to make the whole process complete. Third, it would be interesting to move on the task of report generation for the application in the online self-diagnosis.

References

[1] Chaitanya Shivade, Preethi Raghavan, Eric Fosler-Lussier, Peter J Embi, Noemie Elhadad, Stephen B Johnson, and Albert M Lai. A review of approaches to identifying patient phenotype cohorts using electronic health records. *Journal of the American Medical Informatics Association*, 21(2):221–230, 2013.

[2] Siddhartha R Jonnalagadda, Abhishek K Adupa, Ravi P Garg, Jessica Corona-Cox, and Sanjiv J Shah. Text

mining of the electronic health record: An information extraction approach for automated identification and subphenotyping of hfpef patients for clinical trials. *Journal of cardiovascular translational research*, 10(3):313–321, 2017.

[3] Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014.

[4] Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-Fai Wong, and Xiangying Dai. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 201–207, 2018.

[5] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. *arXiv preprint: arXiv:1901.10623*, 2019.

[6] Ronald Parr and Stuart J Russell. *Reinforcement Learning with hierarchies of machines*. Advances in neural information processing systems, 1998.

[7] Richard S Sutton, Doina Precup, and Satinder Singh. *Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning*. Artificial Intelligence, 1999.

[8] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sund. Hierarchical reinforcement learning for course recommendation in moocs. *Psychology*, 5(4.64):5–65, 2019.

[9] Jiaping Zhang, Tiancheng Zhao, and Zhou Yu. Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 140–150, 2018.

[10] Jun Feng, Minlie Huang, Yijie Zhang, Yang Yang, and Xiaoyan Zhu. Relation mention extraction from noisy data with hierarchical reinforcement learning. *arXiv preprint arXiv:1811.01237*, 2018.

[11] Mohammad Ghavamzadeh. Hierarchical reinforcement in continuous state and multi-agent environments. 2005.

[12] Yu-shao Peng, Kai-Fu Tang, , Hsuan-Tien Lin, and Edward Y Chang. Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis. 2018.

[13] Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *arXiv preprint: arXiv:1604.06057*, 2018.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[15] Hao-Cheng Kao, Kai-Fu Tang, and Edward Y Chang. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. 2018.

[16] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222, 2018.

[17] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. *arXiv preprint arXiv:1811.03925*, 2018.

[18] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint: arXiv:1704.03012*, 2017.

[20] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *arXiv preprint: arXiv:1805.08296*, 2018.

[21] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. 2018.

[22] Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony Jebara. Subgoal discovery for hierarchical dialogue policy learning. *arXiv preprint arXiv:1804.07855*, 2018.

[23] Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In *Proceedings of NIPS Workshop on Deep Reinforcement Learning*, 2016.