Automated Medical History Taking in Primary Care. A Reinforcement Learning Approach.

Version of May 23, 2022

Zhuoran Guo

Automated Medical History Taking in Primary Care. A Reinforcement Learning Approach.

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

by

Zhuoran Guo
born in Henan, china

Accelerated Big Data Systems Research Group
Department of Computer Engineering
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Automated Medical History Taking in Primary Care.
A Reinforcement Learning Approach.

Author: Zhuoran Guo
Student id: 5246725

Abstract

Online searching for healthcare information has gradually become a widely used internet case. Suppose a patient suffers the symptom but is unsure of the action he needs to take, a self-diagnosis tool can help the patient identify the possible conditions and whether this patient needs to seek immediate medical help. However, the accuracy and quality of the service provided by those self-diagnosis tools are still disappointing and need further improvement. This thesis focuses on an automatic differential diagnosis task with a comprehensive evaluation of reinforcement learning methods. Also, we present a systematic method to simulate medically correct patients records, which integrates a standard symptom modeling approach called NLICE. In this way, we can bridge the gap between limited available patients records and data-driven healthcare methodologies. This project investigates both flat-RL methods and hierarchical RL in an automatic differential diagnosis setting and evaluates the performance of those two kinds of methods on simulated patients records. More specifically, the action space for the differential diagnosis task is inevitably large, so the flat-RL performs relatively poorly in complicated scenarios. The hierarchical RL method can split a complex diagnosis task into smaller tasks: it contains two-level of policy learning, and each low-layer policy imitates one medical specialty. Therefore hierarchical RL method increases the Top 1 success rate from 23.1% in flat-RL method to 45.4%.

Besides the advanced policy learning strategy, this thesis explores the ability of NLICE symptom modeling in distinguishing conditions that share similar symptoms. The experimental results experience increases in flat-RL and hierarchical RL models and finally achieve 36.2% and 71.8% Top 1 success rates, respectively. To further solve the sparse action space problem in the automatic diagnosis domain, the reward shaping algorithm is implemented in the reward configuration part. The average gained reward of hierarchical RL increases from -3.65 to 0.87. Additionally, we model the general demographic background of patients and utilize contextual information to perform the policy transformation strategy, which eliminates the miss classification problem in highly sex-age related diseases.

Thesis Committee:

University supervisor:	Dr. ir. Z. Al-Ars, Faculty EEMCS, TU Delft
Committee Member:	Dr. M. Kitsak, Faculty EEMCS, TU Delft
Company supervisor:	Dr. T. Jaber, Medvice

Preface

During the progress of the completion of this thesis, I have received much guidance, help and support from many people. Here I'd like to express my sincere gratitude to all of you.

Firstly, I am greatly indebted to my supervisor Dr.Zaid Al-Ars for his enlightening guidance through the entire thesis process. Without his great patience and constant encouragement, the accomplishment of this thesis wouldn't have been possible. I really enjoy the time when I share positive results with him. His strong sense of responsibility and professional dedication not only let me know how to drive deeply in academic research but also set a role model for my future career. This thesis could not be finished without the help of Dr.Tareq from Medvice, the feedback he provides me is always the source of ideas, and innovation for this thesis.

Secondly, many thanks go to all the friends who also spend study time at TU Delft. We share happiness and sadness together, we knew each other when we are fresh undergraduate students, but I am happy that all of us grew professional experts in our areas. And my heartfelt appreciation goes to my boyfriend Chenxu Ma, you cheer me up when I was down, and helped me know who I am when I found myself at a loss.

Last but not the least, I'd like to give my special thanks to my beloved parents. Because of their unconditional support, from mental to financial, I could fly higher and explore more in my life. Though we are forced to be separated during covid-19, I know you are always around me and we finally will be together in the future.

Zhuoran Guo
Delft, the Netherlands
May 23, 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Medical history taking, or “anamnesis,” is a fundamental diagnostic procedure of the medical consultation, illustrating the source and reason for a health issue. In almost 70 to 80 percent [4] of all primary health cases, a valid diagnosis can be made based on the medical history of the patient alone. However, this essential step toward making an accurate diagnosis is one of the most complicated and time-consuming tasks a doctor must undertake. A medical consult usually allocates more than 50 percent of the time spent on the medical history-taking process.

With the advent of recent pandemic concerns, population aging, and increased use of healthcare services, the primary healthcare sector faces unprecedented workplace pressure. This growing issue calls for expert solutions that can unburden global primary healthcare systems and for focus on increasing cost-effectiveness and clinical workflow efficiency. For instance, the survey reveals that in 2012 [18], 35% of adults in the U.S. had ever tried an online symptom checker or other medical interaction systems to accomplish self-diagnosis for their diseases. They usually begin by searching a symptom in a search engine like google and gain the diagnosis recommendation results. Even though fast accessibility and no cost are advantages of the online-engine automatic diagnosis, the quality and accuracy still are disappointing. In addition, some online symptom checkers still suffer from insufficient gathering of demographic information from patients [19], which limits the diagnosis accuracy.

One promising method to achieve such automation is to explore data-driven reinforcement learning models to automate the medical history-taking process. In collaboration with Medvice - a Dutch Healthcare Startup which focuses on automating as much of the interaction between patient and doctor as possible - this project explores the design of such an automatic diagnostic system, which could perform a fully autonomous medical history-taking session. This session is also called anamnesis, a process to gather relevant medical symptom information from a specific complaint.

This project aims to develop a “dialogue” system for the automatic medical diagnosis that converses with patients to collect related symptoms and automatically makes a diagnosis. During the interaction process between the patient and the Anamnesis AI interface, our interface will interact with a patient, ask questions about their symptoms based on the interaction results, and then attempt to create a differential diagnosis.

1.2 Problem statement and research questions

As the motivation above illustrates, due to their several disadvantages, healthcare online symptom checkers are still not able to satisfy the patients' needs and would result in incorrect differential diagnoses in many cases. As a result, the main research question we investigated in this thesis could be summarized as: what reinforcement learning algorithm could achieve high efficiency and accuracy of clinical workflows, reduce clinician workload, manage finite resources and help patients access the most appropriate care pathway more rapidly.

Another challenge in the current online symptom checker system is obtaining reliable and realistic patient records due to privacy concerns and copyright limitations. Even though there are databases that are built based on symptom-disease pair probability relationships, these are still insufficient to reflect the symptom representation in the real world. In addition, the symptom-disease pair probability is not always medically correct and does not have medical experts' validation. Another reason is the insufficient expression of symptoms. More specifically, the same symptom may appear in various diseases, but the characteristics of such symptoms vary with multiple factors, e.g., age, gender, frequency, and effect on specific body parts. Lack of symptom characteristics leads to differential diseases that share the same symptom fails. Therefore, this project will also address building a reliable and professional medical database to improve differential diagnosis.

1.3 Thesis contributions

Firstly, this thesis investigates a feasible methodological approach to applying reinforcement learning models using simulated historical training data. From formulating the automatic diagnosis problem to developing an automatic diagnosis chatbox, this project will design a feasible symptom chatbox prototype with the purpose of assisting the medical professional and focusing on developing data-driven reinforcement learning models to automate the medial history-taking process dynamically.

Secondly, to bridge the gap between limited available patient data and data-driven methodologies, we propose an approach to generate synthetic clinical data. As discussed before, distinguishing diseases by only relying on symptoms-diseases probability is not reliable; we enrich the medical database with extra information in collaboration with medical experts in Medvice. In addition, we created a symptom modeling standard approach and named it NLICE modeling, which contains nature, duration, what situations excite the symptom, etc. We could provide a feasible, reliable, and medically correct patient database for further research in the automatic diagnosis area.

1.4 Outline

The thesis is organized as follows. Firstly, we discuss the objective of this thesis in Chapter 1. Chapter 2 introduces current data-driven decision support systems that have been implemented to facilitate medical doctors in the automatic diagnosis domain. The fundamental concepts of reinforcement learning algorithms are also included in this chapter as a reference. We discuss the detailed synthetic data generation procedure in Chapter 3. In Chapter 4, an in-depth explanation is provided for our solution including the methodology and implementation approach, followed by the experimental results and discussion in Chapter 5. Finally, Chapter 6 concludes the thesis and lists future research directions.

Chapter 2

Automatic diagnosis via reinforcement learning

2.1 Background

In the healthcare domain, differential diagnosis is a dynamic interaction between patients and doctors. Doctors will generally analyze patient information such as symptoms and demographic information (medical history, age, gender, etc.) to identify patients' possible diseases and then differentiate them from other cases that may share similar situations. In most cases, patients usually see a doctor and want to inform doctors of their symptoms by answering questions. Nevertheless, few doctors will directly diagnose based on patients' self-reported records. They usually continue to ask more questions or perform clinical tests to make a final diagnosis decision. Usually, gathering those patients' information via asking a series of questions is called the clinical decision process. However, the rules for making diagnosis decisions are still unclear because of the high prevalence of complex diseases and dynamic changes in patients' clinical conditions.

The process of determining a differential diagnosis can be modeled in a number of ways and involves several aspects. One of the first methods proposed for this purpose is the use of decision trees and utility functions, to decide what test to take [10]. These methods have limited accuracy and applicability in real-world scenarios. In order to achieve reasonable prediction accuracy, Bayesian inference [11] was proposed. [12, 11] implement entropy or impurity functions to pick symptoms based on the theory of information gain. To further acquire additional information, analysis of medical imaging and radiology was shown in [14], which firstly accesses the health status of a patient, and may decide to take additional measurements such as diagnostic tests or imaging scans before making a final diagnosis decision. However, these works generally adopt certain greedy or approximation schemes since computing the global maximum of information gain is intractable, and hence this approach results in a reduction in accuracy. Moreover, the problem formulation in these approaches is ambiguous and complex, which limits their generality and applicability. Abstracting differential diagnosis to a simplified problem can more likely be extended to more general cases.

Reinforcement learning [21] is a goal-oriented learning strategy wherein an agent or

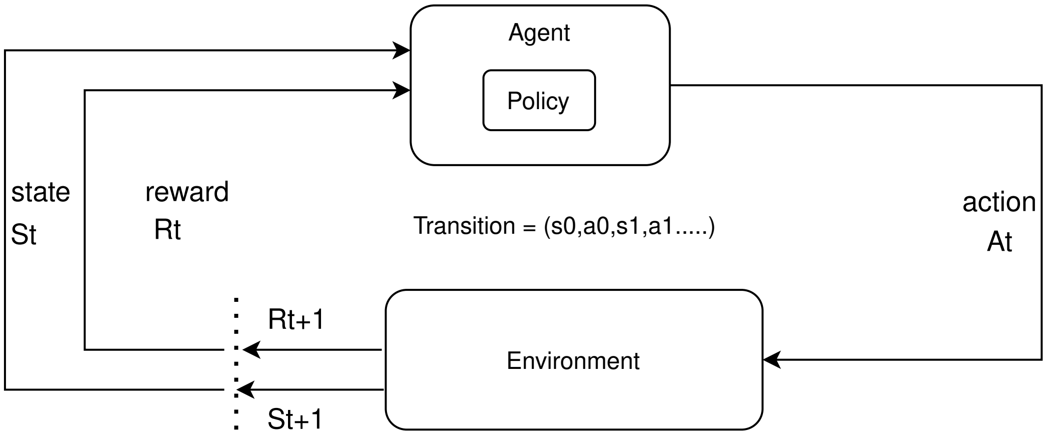

Figure 2.1: RL method for learning by interaction

a decision-maker learns a policy to maximize a reward function by interacting with the environment. Figure 2.1 shows the action-reward loop of a general reinforcement learning model. Deep reinforcement learning has proven to improve the efficiency and to achieve certain human-level accuracy in multiple control domains, such as game playing, computer vision, and autonomous control. An example is Alpha Go, a reinforcement learning agent for playing the strategy board game Go. Given the current state of the Go stones, Alpha Go is able decide where to next place the white/black stone on the board based on its learned optimal policy. Therefore, given the dynamic nature of the automatic differential diagnosis, reinforcement learning (RL) is particularly suitable for symptom checker settings.

2.2 Diagnosis dialogue system as a markov decision process

2.2.1 Terminology

An automatic diagnosis dialogue system is a diagnosis system that requests symptoms automatically from patients, gradually building a symptom profile, then finally makes a decision about a diagnosis. We could formulate this process as Markov Decision Processes (MDPs) [7] and employ reinforcement learning based approaches for policy learning in this system. At the beginning of this section, we will first briefly introduce the basic terms we used in the reinforcement learning approach (see Figure 2.1).

1. Agent: A reinforcement learning agent is the entity trained to make correct decisions.

2. Environment: The environment is the object or surrounding which the agent interacts with.

3. State: The State defines the agent's status at the current timestamp.

4. Action: The choice that the agent makes at current the timestamp based on the current State. We should pre-define the Action Space as a set of actions that an agent can perform.

5. Transition: the function that could calculate the probability of transferring to next State given the current State and Action.

6. Reward Function: the expected reward over all the possible states that one can transition to from State s.

7. Policy: Policy is the algorithm behind an agent picking an action. In practice, it is a probability distribution assigned to the set of actions. Highly rewarding actions will have a high probability and vice versa.

2.2.2 Markov decision process (MDP)

RL is associated with the more difficult setting in which no prior knowledge about the Markov decision process is presented. Therefore the basic definition of MDP can be found as follows.

At each step, an RL agent obtains evaluative feedback about the its action, allowing it to improve the performance of succeeding actions. Generally, Markov Decision Process is defined by

$$(S,A,P,R,\gamma) \tag{2.1}$$

where S is States, A is set of Actions, P represents the probability transition function, R means reward, γ means discount factor.

By applying action $a \in A$ in a state$s \in S$, the system makes a transition from s to a new state $s_0 \in S$, based on a probability distribution over the set of possible transitions [23]. The probability ending up in state s_0 after doing the action, a state s is denoted by T(s, a, s_0).

$$P(s_{t+1}|s_t) = P_t(s_{t+1}|s_1...s_t) = T(s_t, a_t, s_{t+1}) \tag{2.2}$$

which means that the future state at timestamp t+1 only depends on present state at timestamp t.

In Markov Decision Process, Transition is defined as follows:

$$P^a_{ss'} = P[s_{t+1} = s'|s_t = s, a_t = a] \tag{2.3}$$

and Reward Function is defined as:

$$R^a_S = E[r_{t+1}|s_t = s, a_t = a] \tag{2.4}$$

The state reward R^a_s is the expected reward in all the possible states that one can transit to from state s. Furthermore, according to the state transition probability and reward function defined before, the rewards and the next state also depend on the agent's action in the earlier timestamp.

As mentioned before, a policy function defines the algorithm behind making a decision or picking an action. It also illustrates the behind behavior of an RL agent. Formally, a policy is a probability distribution over the set of actions a, given the current state s, i.e., it gives the probability of picking an action at state s. Moreover, policy function is defined as:

$$\pi(a|s) = E[r_{t+1}|a_t = a, s_t = s] \tag{2.5}$$

The MDP is solved to get an optimal policy π. A policy π is defined as $\pi(s) \to a$, which is a mapping from states to actions that typically represents the behavior of the agent. π^* is an optimal policy that maximizes the cumulative reward at the end of an interaction session.

2.2.3 Automatic diagnosis system formulation

In this chapter, we define a diagnosis dialogue system as progress that tries to achieve as much as high differential diagnosis accuracy efficiently through a series of interactions with the user. We show that by quantifying the terms efficiency and achievement of application goal in terms of an objective function, the dialogue system can be described as a known stochastic model MDP that can be used for learning the differential diagnosis strategy for a given patient's state.

Moreover, before we formulate the automatic diagnosis system, some notations should be first built. We use I and D to represent the set of all symptoms and all diseases. In the diagnosis context, the way that the agent interacts with a patient could be found as follows: at the beginning of the dialogue session, the agent is provided with a symptom that the patient has from the set of all symptoms I. This initial symptom is regarded as the main complaint and provided by the patient. The agent picks a symptom $i \in I$ to ask the patient at each round step. Then this patient responds to the agent with a Yes or No answer indicating whether he/she is suffering from that particular symptom. At the end of the diagnosis interaction, the agent predicts a disease that the patient may have from all sets of all diseases D. In reinforcement learning concepts, the agent's goal is to maximize the expected reward during the interaction process. Therefore the final goal of automatic diagnosis system is to correctly predict diseases and inquire as much as possible correct symptoms the user may suffer.

We also formulate the state space and action space as follows:

1. State: A state s contains symptoms requested by the agent and informed by the user till the current time t, the earlier response of the user, the earlier action of the agent, and the step information. Regarding the representation of symptoms, its dimension is equal to the number of sets of symptoms, whose elements for positive symptoms are 1, negative symptoms are -1, not sure symptoms are -2, and not mentioned are 0 [13]. Each state $s \in S$ is the concatenation of these three vectors.

2. Action: In time step t, the agent receives a state S_t and then selects an action from a discrete action set A according to a policy . In our formulation, $A = I \cup D$.

The goal for the agent is to find an optimal policy so that it can maximizes the expected cumulative future discounted rewards

$$R_t = \sum_{t'=t}^{T} {}^{t'-t} r_t' \tag{2.6}$$

where $\gamma \in [0,1]$ is the discounted factor and T is maximal turn of current dialogue session. The Q-value function

$$Q^{\pi}(a,s) = E[r_t | a_t = a, s_t = s, \pi] \tag{2.7}$$

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

The optimal Q-value function is the maximum Q-value over all possible policies:

$$Q*(s,a) = max_{\pi} Q^{\pi}(s,a) \tag{2.8}$$

It follows the Bellman equation:

$$Q^{*}(a,s) = E_{s'}[r + \gamma max Q^{*}(s',a')|s,a] \tag{2.9}$$

where $a' \in$ A, policy π is optimal if and only if for every state and action,

$$Q^{\pi}(s,a) = Q^{*}(s,a) \tag{2.10}$$

Then the policy can be reduced deterministically by

$$\pi(a|s) = argmax_{a \in A} Q*(s,a) \tag{2.11}$$

2.3 Reinforcement learning based methods

The main challenge of the automatic diagnosis task is how to gather the underlying dynamics and uncertainties of the reasoning process, then inquire about accurate symptoms under limited labeled data and within limited interaction turns. Most current methods usually address this problem as a sequential decision-making process, then formulate the process as Markov Decision Processes and employ Reinforcement Learning for policy learning. RL learns what is the next symptom should be asked and make a disease diagnosis at the end of session, which partly deviates from the doctor's diagnostic procedure. This chapter lists some state-of-the-art reinforcement learning models for automatic disease diagnosis.

2.3.1 Policy learning with Deep-Q network

DeepMind firstly developed the DQN (Deep Q-Network) algorithm in 2015 [16]. DQN could solve a wide range of Atari games (some to superhuman level) by combining reinforcement learning and deep neural networks at scale. Before the DQN was proposed, q-learning was the most mainstream reinforcement learning algorithm, which used a table to store the state function and action function [25]. The basic idea behind q-learning is to estimate the action-value function by using the Bellman equation as an iterative update,

$$Q_{i+1}(s,a) = E[r + \gamma \underset{a'}{max} Q_i(s',a')|s,a] \tag{2.12}$$

Such value iteration algorithms converge to the optimal action-value function, $Q_i \rightarrow Q*$ as $i \rightarrow \infty$ [20]. However, this basic approach is impractical in practice because the action-value function is estimated separately for each sequence without any generalization. In other words, representing the Q-function as a table containing values for each combination of s and a is impossible for the automatic diagnosis task. Rather, it is typical to use a function approximator to estimate the action-value function $Q(s,a;\theta) \approx Q*(s,a)$, where θ is a parameter included in neural network. DQN enhances the q-learning algorithm by replacing the value-state table with a neural network and experience replay technique.

The function approximator could be done via the minimizing the following loss at each step t :

$$L_i(\theta_i) = E_{s,a,r,s' \sim \rho(.)}[(y_i - Q(s,a;\theta_i)^2] \tag{2.13}$$

where $y_i = \gamma max_{a'} Q(s,a;\theta_{i-1})$. the parameters from the previous iteration θ_{t-1} are fixed and not updated. We use a "target network" to copy the network parameters from a few iterations back instead of the latest iteration.

To avoid calculating the full expectation in the DQN loss, we can minimize the loss using stochastic gradient descent. If the loss is computed using just the latest transition s,a,r,s', this will be reduced to standard Q-Learning.

Besides target network techniques, Atari DQN work [16] proposed a new technique called Experience Replay to update the network parameters stably. The transitions are added to the replay buffer at each data collection step. Then during the training procedure, not use just the latest transition to calculate the gradients and weights. They compute them using a mini-batch of transitions sampled from the replay buffer. This could contribute to two main advantages: higher data efficiency by reusing each transition in many updates and improving stability using uncorrelated transitions in a batch.

2.3.2 Flat-RL based method

A flat RL Agent is trained with a classic flat reinforcement learning method that uses gained rewards to learn a signle-layer dialogue policy. Existing work [22, 26] employs flat reinforcement learning (flat-RL) based methods for the policy learning of the diagnosis system.

The basic idea behind the flat-RL method [26] is discussed next. As Figure 2.2 shows, there are two entities in the system: agent and patient. The agent is a DQN network and works as a doctor. The patient works as an environment and interacts with the agent. The patient's feedback usually contains diagnosis response and inquiry response, which depends on the type of action the agent picks. If the agent picks an inquiry action, the patient responds to this inquiry and continues to the next round. Otherwise, the patient is informed of the diagnosis results, and then the whole session is terminated.

At each turn T, the user takes action A_t, according to the current user state S_t, and the previous agent action A_{t-1}, and transits into the next user state S_{t+1}. The user initiates every dialogue session via the user action A_u, consisting of the requested diseases and all symptoms. As for symptoms requested by the agent during the dialogue, the user will take one of the three types of actions, including True (the symptom appears), False (the symptom does not appear), and not sure (the symptom is not known) after inquiring about a certain number of symptoms. Then the agent picks an action that belongs to diseases as the final diagnosis. The reward will be positive if the agent predicts the right disease or the symptom appears in the user's true symptoms. Meanwhile, the correct predicted disease status means that this dialogue session will be terminated as successful by the patient. Otherwise, the

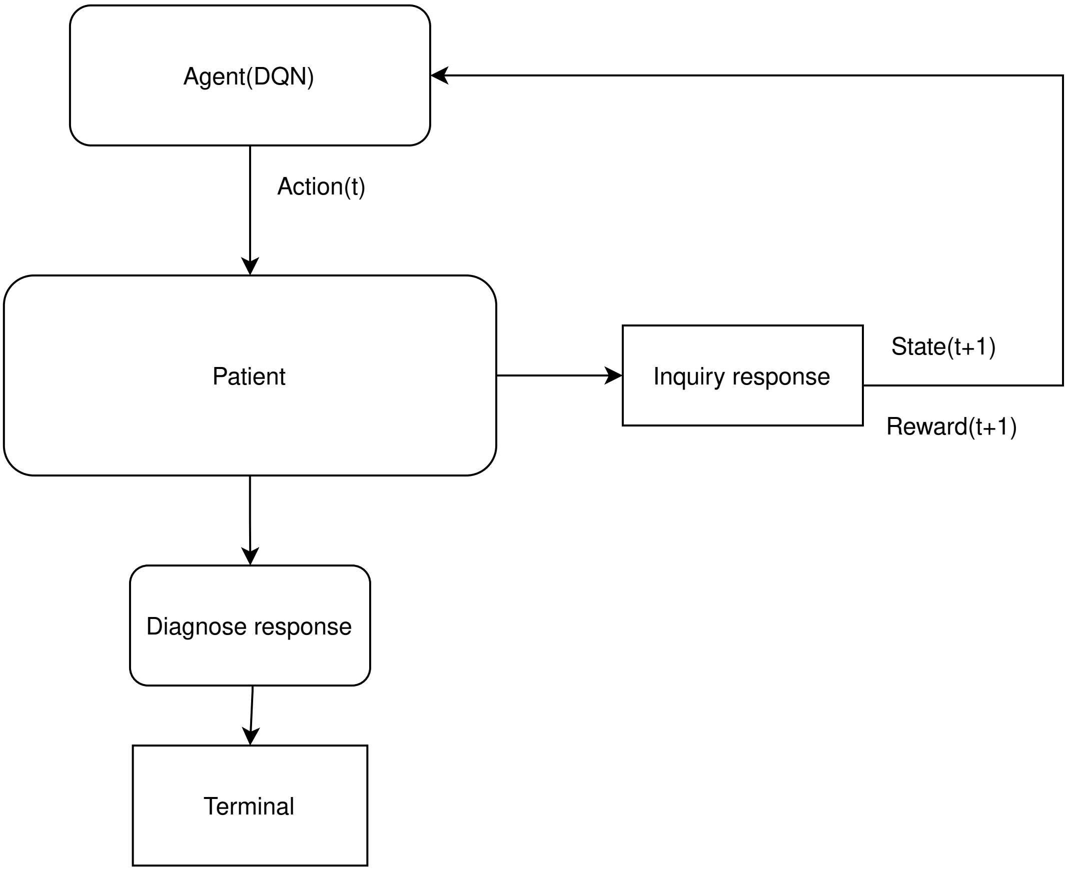

Figure 2.2: Flat-RL based method architecture [26]

dialogue session will be treated as failed if the agent makes an incorrect diagnosis or the dialogue turn reaches the maximum dialogue turn T [26].

The flat-RL method employs a deep-Q network [16] for parameterizing the policy, which takes the state s_t as input and outputs $Q(s_t, a|\theta_i)$ for all actions a. A Q-network can be trained by updating the parameters θ_i at iteration i to reduce the mean squared error between the Q-value computed from the current network $Q(s_t, a|\theta_i)$ and the Q-value obtained from the Bellman equation $y_i = r + \gamma \underset{a'}{max} Q(s_0, a_0|\theta_i)$, where $Q(s_0, a_0|\theta_i)$is the target network with parameters θ_i from some previous iterations.

Even though the performance result in [26] shows that it could achieve 57 percent accuracy in the self-constructed dataset for diagnosis, this accuracy level could not reflect the diagnostic procedures in the real world. One reason is that the task-orientated dataset used for evaluation only contains four types of diseases. Therefore, it only indicates that the flat-RL method could be reasonable for simple tasks with only small action space. Besides the lack of large-scale dataset evaluation, the interaction way could not represent the doctor's hospital behavior.

More comprehensive experiments on large-scale synthetic clinical data could be found in another flat-RL method [22], which proposes flat-RL models in an anatomical manner to simulate real doctors who belong to different departments in hospitals. Anatomical-Rl method creates model to be an ensemble model of various anatomical parts: $M = \{m_p | p \in P\}$. They selected eleven representative body parts:

$$P = \{abdomen, arm, back, buttock, chest, general\ symptoms, head, leg, neck, pelvis, skin\} \tag{2.14}$$

The model m_p of each anatomical part $p \in P$ is responsible for a subset of diseases $D_p \in D$. Similarly, it represents the subset of the symptoms associated with m_p by $I_p \in I$. DQN is selected as the Q-value estimator for each anatomical part model m_p. The action set of m_p is $A_p = I_p \cup D_p$. The state s_p for m_p is a combination of related symptom statuses. Then they trained those eleven models m_p simultaneously, and each trained model was in charge of an anatomical part p.

When a patient asks about an initial symptom i to the anatomical-Rl system, it will select the representative model m_{repr} that has the highest accuracy for initial symptom i by using SymptomModelMapping algorithm [22]. The actual inquiry and diagnosis process is performed by the m_{repr}. This concept is similar to the real hospital, where they regard different anatomical parts as several experts, and then these experts jointly perform practical inquiries and diagnoses.

2.3.3 Hierarchical reinforcement learning

However, existing flat-RL approaches to automatic diagnosis problem typically work well in simple tasks but have significant challenges in complex scenarios. Although flat-RL methods approaches have shown positive results for symptom acquisition [26], when it comes to hundreds of diseases in the real-world, flat-RL policy learning setting is impractical. Furthermore, anatomical-Rl [22] method outperforms flat-RL in large-scale synthetic clinical data, the performance of anatomical-Rl method is still relatively disappointing, especially in the dataset that contains more than 50 diseases. Besides that, several issues could also be found in anatomical-Rl diagnosis procedure: For example, it is possible that the target disease does not belong to the disease set of the chosen anatomical part. Also, other anatomical parts are not fully utilized because it only chooses a single anatomical part to interact with users.

In order to solve the sparse action space problem existing in flat-RL methods, Hierarchical Reinforcement Learning (HRL) [17], in which multiple layers of policies are trained to perform decision making, has been successfully applied to automatic diagnosis tasks. The main idea in the currently proposed approaches [13, 9] is to perform a group of doctors with different expertise who can jointly diagnose a patient. A group of doctors is called low-layer agents. Each low-layer agent has a different action space as it only represents diseases and symptoms related to its expertise. Since a patient can only accept an inquiry from one doctor at a time, a host, called master agent in HRL terms, is in charge of appointing doctors of different expertise to inquire about the patient. As Figure 2.3 shows, at each step, the master agent is responsible for appointing a lower-layer model to perform a

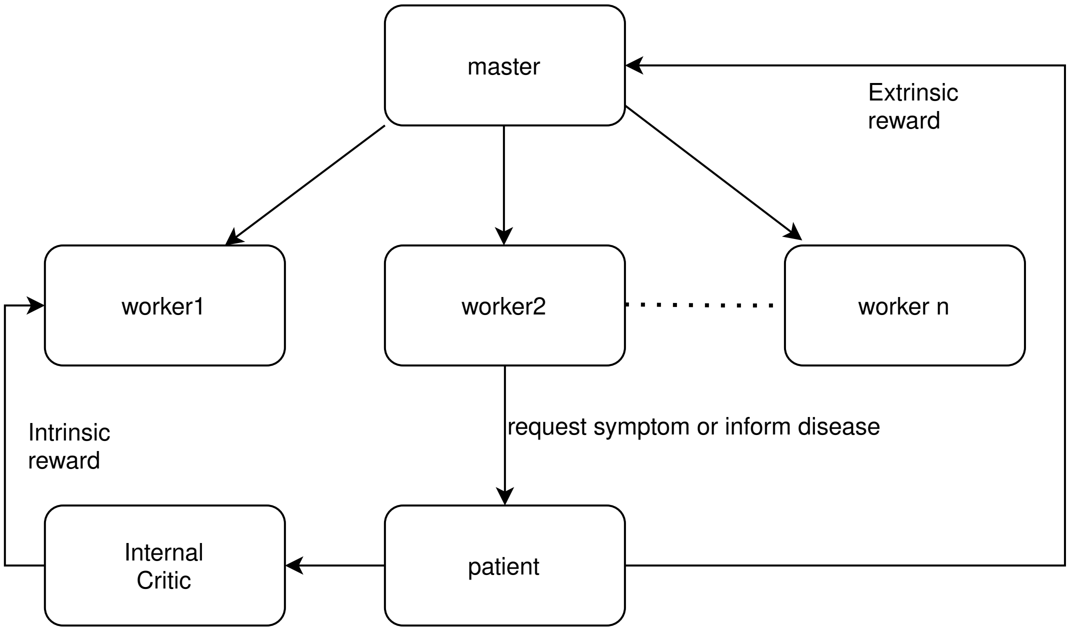

Figure 2.3: HRL-based method architecture[13]

symptom inquiry or a disease prediction. This approach essentially creates a hierarchy in the HRL model by introducing a level of abstraction since the master agent cannot directly perform inquiry and prediction actions. Moreover, several low layer works perform direct interaction with the patient. Like the flat-RL system, the patient in the HRL method works as an environment that interacts with agents. Internal critic in the HRL method defines the reward configuration for each response given by the patient, and it will generate an intrinsic reward for the current lower layer worker. Extrinsic reward is the accumulation of intrinsic rewards from lower layer works.

Usually, hrl-based methods group all the diseases in D into h subsets $D_1, D_2, ..., D_h$, where $D_1 \cup D_2 \cup \cdots \cup D_h = D$ and $Di \cap D_j =$ for any $i \neq j$ and $i, j = 1, 2, \cdots, h$. Each D_i is associated with a set of symptoms $S_i \in S$, whose symptoms related to diseases in D_i. While worker w_i is responsible or collecting information from user about symptoms of Si or inform disease.Hierarchical Reinforcement Learning enables autonomous decomposition of challenging long-horizon decision-making tasks into simpler subtasks, then reducing action space when we have hundreds of diseases and symptoms.

2.4 Evaluation matrix

There are several famous metrics to evaluate the performance of the diagnosis system, including success rate, average reward. That evaluation matrix could determine how effective the trained agent is and perform as an approach for comparing different models on the same task.

2.4.1 Top n success rate

$$Success\ rate = \frac{Number\ of\ SD}{Number\ of\ samples} \tag{2.15}$$

where SD means successful diagnosis. Furthermore, generally, three kinds of conditions could indicate that the dialogue is successful. Top 1 success means diagnosis ends with a successful inform user correct Top 1 disease, and Top 3 or Top 5 success means that correct disease is included in Top 3, or Top 5 informed diseases.

2.4.2 Average turn

$$Average\ Turn = \frac{\sum_i^N L_i}{Number\ of\ samples} \tag{2.16}$$

where N is equal to several episodes, L_i means the number of turns in $episodes_i$. Average Turn could evaluate the average number of symptoms inquired by an agent. More symptoms inquiries in an episode play a positive role in disease prediction. However, it will not be user-friendly because patients usually do not have the patience to answer long series of questions.

2.4.3 Matching rate

Matching rate means the proportion of dialogue requested about the at least one correct symptom that the user indeed occurs.

$$Matching\ Rate = \frac{Number\ of\ matching\ samples}{Number\ of\ samples} \tag{2.17}$$

2.4.4 Average reward

Namely, average reward is the average rewards that agent gained from environment.

$$Average\ Reward = \frac{\sum_i^N R_i}{Number\ of\ samples} \tag{2.18}$$

2.4.5 Error confusion matrix

Error Confusion Matrix consists of multiple squares with two dimensions and is widely used to evaluate classification problems. In the automatic diagnosis task, the square in an error confusion matrix with true class i and predicted class j means a disease in medical specialty group i is mis-classified into medical specialty group j by the disease classifier.

Chapter 3

Synthetic data generation

3.1 Symptom modeling: an NLICE approach

Due to privacy concerns and laws (e.g., the Health Insurance Portability and Accountability Act; HIPAA), the clinical records from the real world may not be released publicly. Even though current available medical datasets have been used for medical research domains, like the MIMIC-III [8] and PCORnet [15], the scope is only limited to real-time interpretation of data in the ICU setting. Thus, the current medical datasets lack the relationship information between symptoms and conditions. These limitations in current medical datasets have led several research efforts to resort to synthetically generated patient records. For example, using data from publicly available medical databases such as PubMed [2], other sources such as medical literature, published records of disease symptom relationships, etc. Symcat, the most popular symptom-disease database, is widely used in the automatic diagnosis task. Since each disease in Symcat is associated with a set of symptoms and probabilities, i.e.,$p(s|d)$. An example of disease-symptoms probabilities pair could be found in Table 3.1. Besides relationships between symptoms and conditions, Symcat also includes the preva-

Condition	Associated Symptoms	Probability
abscess-of-the-pharynx	sore-throat	83%
	fever	69%
	nasal-congestion	58%
	cough	51%
	difficulty-in-swallowing	48%
	ear-pain	34%
	headache	34%
	sharp-chest-pain	27%
	coughing-up-sputum	27%
	allergic-reaction	20%

Table 3.1: Selected sample disease of Symcat

Condition	Age distribution	Probability
abscess-of-the-pharynx	<1 years	8.9%
	1-4-years	27.8%
	5-14-years	22.2%
	15-29-years	13.3%
	30-44 years	10%
	45-59 years	10%
	60-74 years	3.3%
	75+ years	4.4%

Table 3.2: Age distribution of a selected disease

lence of diseases and symptoms by age, gender, race, and ethnicity. An example of the age distribution for abscess of the pharynx disease could be found in Table 3.2. that additional relationship information could also be helpful to diagnosis correctly for diseases that share similar symptoms. For example, there are several types of diabetes, but juvenile diabetes typically appears in people who are under the age of 30, whereas Type 2 diabetes is more common after age 45 [3].

Even though utilizing context information(age, gender) of a patient in a diagnosis procedure could mitigate the difficulty of identifying diseases that share similar symptoms, the diagnosis procedure still heavily relies on symptoms appearing in a patient. During the project, one thing that became apparent is that a true/false binary modeling of symptoms is not enough to make proper distinctions, especially between diseases that showed similar symptoms. To address this problem, an alternative approach to Symptom Modeling is explored.

3.1.1 Symptom data introduction

For most automatic diagnosis tasks, symptoms were modeled as binary values - a value of 1 representing symptom presentation and 0 denoting an absence of the symptom. This approach to modeling the symptoms was motivated by the nature of the data available in Symcat. However, more information can be obtained about a symptom without having to perform any laboratory or diagnostic tests. This approach to symptom inquiry and the potential for improved performance over the more straightforward yes-no modeling approach is the focus of this section.

3.1.2 NLICE description

When a particular symptom is expressed, the current symptom model marks the symptom as being present (i.e., having a value of a positive integer). If it is absent, the value is negative or zero. However, other characteristics of the symptom's expression might help make a differential diagnosis in practice. These characteristics are summarized with the NLICE acronym, which is short for Nature, Location, Intensity, Chronology, and Excitation.

Synthea's expressive generic module framework meant that as long as statistical relationships between these additional symptom descriptors and respective conditions were present, realistic data could also be generated.

Nature

It is also possible to obtain the nature of the presented symptom. As an example, consider cough, which is a common symptom associated with respiratory infections. A 1 or 0 approach to modeling this symptom ignores the different ways cough might present. Some patients might experience a dry cough. Others might have a cough that produces mucus, etc. These differences define the nature of the cough and provide information that might make it easier to distinguish between diseases that have similar symptoms.

Location

The location of a symptom can also be a discriminating factor when making distinctions between diseases. As an illustration, consider a patient who complains of abdominal pain. The abdomen can be separated into four parts in medical anatomy: upper and lower right quadrants and upper and lower left quadrants. There is also a division into nine regions: right and left hypochondriac, right and left lumbar, right and left iliac, and the epigastric, umbilical hypogastric regions [6]. Additional information on which regions/quadrants in the abdomen are affected by this pain might help make easier recognition regarding the causal disease.

Intensity

The symptom modeling should also have the ability to accurately evaluate the intensity of a symptom. Symptom intensity could represent how severe the symptom presentation is. It can also be a pointer to the causal disease. For example, pain is a typical ”experience” in many symptoms. A patient suffering from appendicitis is likely to experience severe to moderate abdominal pain.

One remark is that intensity is usually a uniquely personal experience. Sometimes it is hard to quantify intensity as several factors accurately could heavily affect human experiences, like a patient's emotional and mental state [5].

Chronology

Chronology can be divided into three subterms: the first one is frequency, representing how often the symptom happens. Furthermore, duration illustrate how long the symptom persists, and finally onset is when the patient started experiencing this symptom. These could also be additional factors when making a differential diagnosis.

Condition Group	Condition Name	Condition Group	Condition Name
Pulmonary	Covid	Neurological	Migraine
	Seasonal influenza		Tension headache
	Pneumococcal pneumonia		Subarchnoidal bleeding
	COPD		Radicular syndrome
	Asthma		Lumbago
	Pneumothorax		Cauda equina
	Pulmonary edema		Lumbar spinal stenosis
Gastrointestinal	Salmonellosis		Carpal tunnel syndrome
	Yersinosis		Bacterial meningitis
	Giardinasis		Viral meningitis, enterovirus
	Norovirus		Viral meningitis, varicella zostervirus
	Appendicitis		
	Functional constipation		
	Acute pancreatitis		
	IBS (Constipation)		
	Atrophic gastritis		

Table 3.3: Selected conditions for NLICE Analysis

Excitation

Excitation refers to activities or scenarios that the patient performs or is exposed to that activate or worsen the symptoms. For example, a patient might complain of heart pain only while swimming. If we captured excitation information, we could also make for a more accurate distinction between diseases.

3.2 Data collection

As mentioned before, the data supplied in Symcat was not immediately suited to NLICE. modeling approach. Some symptoms present in Symcat did capture some components of this approach e.g., some conditions presented with *burning-abdominal* pain and others with *sharp-abdominal* pain (both distinctions made on the nature of the abdominal pain). However, this was not available for all the symptoms supporting the NLICE approach. Hence, as a proof of concept, data for a small set of conditions were gathered from medical literature. This data-gathering effort was carried out by medical experts from our industry collaborators at Medvice. The conditions were restricted to ten categories for the data collection based on the medical specialty. The selected three groups of conditions are shown in Table 3.3

For each of these conditions, data regarding typical symptoms, associated expression probability, and NLICE characteristics were extracted manually from medical literature.

Condition	Symptom	Probability	NLICE	Probability
covid-19	Cough	68%	Nature: Mucus	33%
	Altered breathing	47%	Nature: Dyspnea	100%
	Fatigue	38%	N/A	
	Pain	14%	Location:Head	99.81%
			Location: Chest	0.11%
	Fever	43.8%	Intensity: Mild	78.2%
			Intensity: Moderate	18.2%
			Intensity: Severe	3.5%
	Soreness	14%	Location: Throat	100%
	Vomiting	5%	N/A	
	Congestion	48%	Location: Nose	100%
	Discoloration	0.13%	Nature:Blue (Cyanosis)	100%

Table 3.4: NLICE symptoms modeling for covid-19

Condition	Age distribution	Probability	Gender distribution	Probability
covid-19	0 - 14 years	1%	male	42%
	15 - 49 years	55%	female	58%
	50 - 64 years	30%		
	>65 years	15%		

Table 3.5: Contextual information modeling for covid-19

One detailed example could be found in Table 3.4 and 3.5.

Using the generation method described in Section 3.3 NLICE datasets were generated based on the modeling symptoms with NLICE approach. And in total we group those conditions via medical specialty, the conditions and symptoms statistics is present in Table 3.6. It should be stated that symptoms do not have to support all NLICE characteristics. For example, fever can be characterized by its intensity, but it would not be logical to associate a location with fever.

3.3 Synthetic patient records generation

The synthetic patient records generation approach is an extension of the work [1]. Synthea is a Synthetic Patient Population Simulator that could export synthetic realistic (but not accurate) patient health records in various formats. As Synthea source code is public, we could easily modify it to suit the target tasks.

The exact architecture of Synthea can be found in Figure 3.1. Clinical care maps and incidence & prevalence statistics are in charge of building patients' models, which will be used later for disease progression in disease modules. Disease modules can encode disease models as state transition machines in JSON format. In addition, census data demographics

medical speciality	Num. of diseases	Num. of symptoms
Pulmonary	10	18
Gastrointestinal	9	13
Neurological	4	19
Neuro-orthopaedic	6	20
Rheumatological	5	6
Urological	3	19
Oropharyngeal	3	13
STDs	7	28
Oncological	4	16
Deficiencies	4	6

Table 3.6: Overview of the NLICE-Modeling data

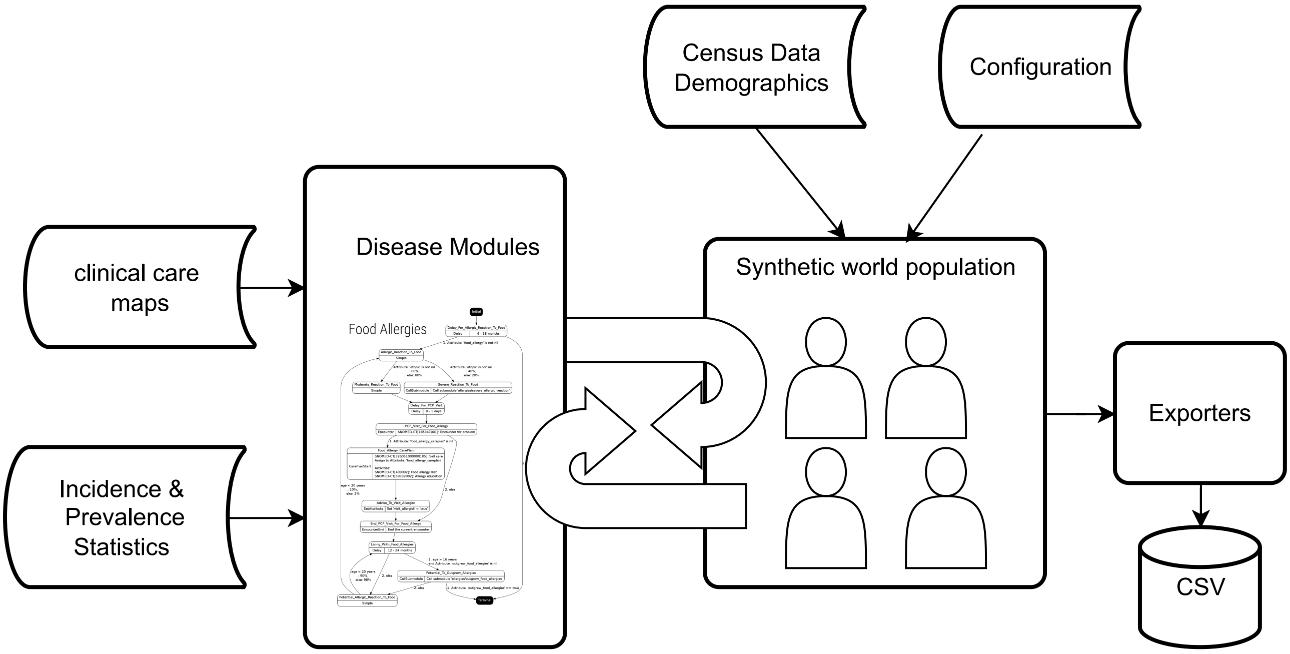

Figure 3.1: Synthea's architecture [24]

and configuration seed the synthetic world [24]. Finally, the state and transition in disease modules simulate the synthetic world population and feed the simulation results to the exporters. We choose the CSV format to save the patients' event dataset.

The most critical component in Synthea architecture would be Synthea's Generic Module Framework (GMF), which enables the modeling of various diseases and symptoms that generate the medical history records of synthetic patients. It simulates patients' records independently from birth to the present day. These modules are based on state machines, where each state depicts one stage in the disease progression procedure. There are two general classes of states in Synthea inside: the Clinical States and the Control States.

As the name suggests, clinical states provide medical knowledge when modeling conditions. They assume encounters within health care facilities or seeing doctors, the condition the patient develops, prescribed medications, and observations varying from temperature assignments associated with the condition to conclusions with additional complicated laboratory tests.

On the other hand, control states show the birth and termination of the disease module and introduce pauses between different stages, e.g., a usually time delay could happen between the beginning of medication and the next patient with the doctor. Some control states describe unique characteristics of the patients or correct errors on a particular stage, e.g., It is impossible that male patients to suffer from pregnancy. All states must include a transition procedure. The Terminal state implies the termination of the module, so there is no transition after the terminal state. This property specifies which next state in the patient's module should transfer. Integrated with the clinical and control states, the state transition characterizations, prevalence statistics, and clinical care maps, A patient's interaction with doctors while suffering from various symptoms could be modeled by the Synthea simulator.

In Synthea inside, modules are responsible for modeling clinical events that could appear in a patient's lifetime, describing stages' progress and shifts.Synthea's expressive generic module framework meant that as long as statistical relationships between these additional symptom descriptors in Section 3.1.2 and respective conditions were present, then realistic data could also be generated.

Parsing the original collected conditions-symptoms dataset was also developed with medical experts in Medvice. This function aims to output the Synthea-compatible disease modules based on our custom medical data. Then Synthea could incorporate the probabilistic symptom-condition relationships above in a Synthea module generating process, and modules are formed by real-world statistics collected by Medical experts in Medvice. Synthea modules were generated for all 55 diseases present in the Medvice database. An example generated module for Asthma is shown in Figure 3.2

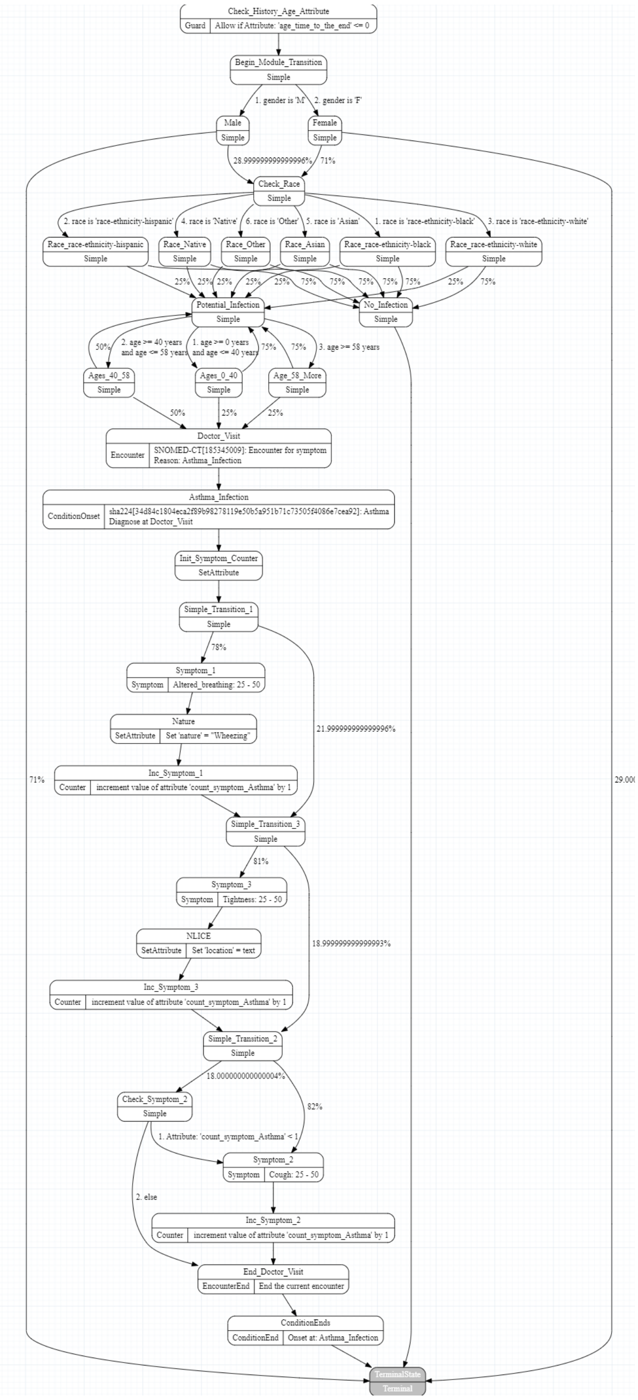

Figure 3.2: Generated Synthea Module for Asthma. Not all states are shown to allow for an easier visualization

Chapter 4

Methodology and approach

4.1 Data processing

Referring to the data generation procedure mentioned in Chapter 3, we finally generated patient records with NLICE-modeling symptoms and contextual information. One sample can be found below.

As the Synthea architecture in Figure 3.1 represents , demographics configuration defines synthetic patients demographics information. For all datasets we generated in this project, we generated the demography of patients based on census data from the state of Massachusetts in the United States of America. Besides areas, we configure the age, gender for each disease in demography based on the contextual information modeling mentioned in Table 3.5.

GENDER	RACE	AGE	PATHOLOGY	SYMPTOMS
F	Asian	30	Acute pancreatitis	Discoloration:Yellow(Jaundice):::::::; Altered appetite:Decreased(anorexia):::::::; Pain:Radiating:::::::; Nausea::::::::; Flatulence::::::::; Bloating::::::::
F	White	48	Rheumatoid arthritis feminine	Disability::::::::; Altered appetite:Decreased(Anorexia):::::::; Fatigue::::::::; Stiffness::::::Morning::; Weight loss::::::::
M	Naive	87	Pulmonary edema oldest	Pain::::::::; Altered mental state:ConfusionORdelirium:::::::; Altered breathing:Dyspnea:::::::

Table 4.1: Sample Generated Data for Patients with Acute pancreatitis, Rheumatoid arthritis feminine and Pulmonary edema oldest

We export the dataset in CSV format. Each row represents one patient record, including demographics and symptoms information. Sex and Age differences in certain diseases are apparent. The generated synthetic datasets utilize the pre-defined contextual information modeling from medical experts, thus contributing to a reliable and realistic patient dataset. For instance, Rheumatoid arthritis feminine could only appear in women. Thus all patient records with Rheumatoid arthritis feminine are only for females. Similarly, Pulmonary edema oldest is a disease occurring in patients older than 85 years old.

Regarding symptoms in each record, we separate symptoms via a semicolon symbol, and its eight types of NLICE features follow each symptom. $[symptom : nature : location : vas : duration : onset : excitation : frequency]$.Nevertheless, each condition does not need to have all NLICE features. This way aims to be easier to work with people without sacrificing machine usability while being more complete in the depiction of medical conditions. The synthea displays only those conditions for which ”NLICE” is known. E.g., if there is no data about the *nature* of, for example, the fatigue present in covid-19, then the NLICE point ’nature’ is not displayed. For the *administrative* and *epidemiology* data, however, this is not the case since their amount is fixed and relatively small.

4.2 Flat reinforcement learning approach

The first chosen method is flat-RL based method, which only contains one policy layer to train the deep-q network. When choosing actions, both symptoms and diseases are regarded as actions. As the architecture shows in Figure 4.1, main components(User simulator, Agent, Inquiry response, Diagnose response and Terminal) have the same functionalities as the system in Figure 2.2. The database NLICE in Figure 4.1 provides fixed NLICE questions for each symptom. The agent accepts a state that comprises all symptoms status at timestep t. The status is decided by the response of the user when receiving actions. Initially, all symptoms status starts from ”unknown” except initial complaint. Therefore, an appropriate encoding symptoms status should be considered. In general state encoding scheme in flat-RL methods [22], one eight-dimensional vector is utilized for representing status of $symptom_i$. Each symptom will have three status types: unknown, yes, or no. When an agent chooses an action that belongs to symptoms, then this action is treated as an inquiry action. Then the user replies to the agent with a ”yes/no” answer. The next state updates the status of the corresponding symptom and restarts the interaction procedure. Nevertheless, to take advantage of the extra NLICE features we gained, the coding scheme for status should be changed slightly. An eight-dimensional vector should be chosen for representing status of $symptom_i$: For positive symptoms, it will be encoding as $[1, Nature, Onset, Frequency, Intensity, Excitation, Duration, body]$. For negative symptoms, vector $[-1,-1,-1,-1,-1,-1,-1,-1]$ is used. Unknown symptoms are encoded as $[0,0,0,0,0,0,0,0]$.

Then all symptoms vectors are concatenated to form the final status vector

$$s = [b_1^T, b_2^T, ..., b_n^T]^T \tag{4.1}$$

where b_i is status of symptom, and n is the number of total symptoms.

Name	Type	Input Size	Output Size
FC1	Linear Relu	$\|I\| * 8$	$1024 \times \omega$
FC2	Linear Relu	$1024 \times \omega$	$1024 \times\omega$
FC3	Linear Relu	$1024 \times \omega$	$512 \times \omega$
FC4	Linear	$512 \times \omega$	$\|I\| + \|D\|$

Table 4.2: The network architecture of our DQN model

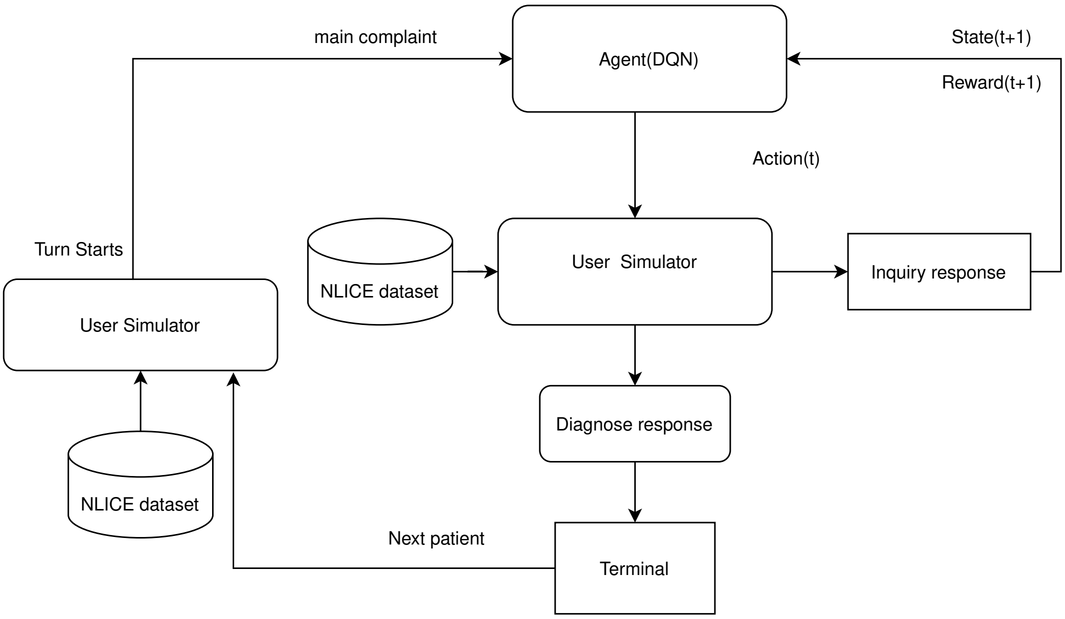

Figure 4.1: Flat-RL system architecture

The notations we used in the flat-RL method include I,D, and A, representing sets of symptoms, diseases, and action space. Given a state vector, our DQN model outputs the Q-value of each action $a \in A$. The detailed architecture of the DQN model can be found below: In our definition, each action consists two types: an inquiry action ($a \in I$) or a diagnosis action ($a \in D$). If the maximum Q-value of the outputs corresponding to a symptom inquiry action, then our model asks about the symptom to the current user, then this user provides feedback before proceeded to the next stage. According to our state encoding scheme, the feedback is incorporated into the next state st+1. Otherwise,the maximum Q-value corresponds to a diagnosis, and a diagnosis action means the termination of the interaction session.

4.3 Hierarchical reinforcement learning framework for disease diagnosis

The main advantages of flat-RL based method are easy development and efficiency for simple disease diagnosis task that includes a limited number of diseases. However, based on the experimental results proposed in [22], the single-layer agent could only inquire around 1 to 3 symptoms questions. These short inquiry sequences indicate that the flat-RL model typically predicts diseases based on the main complaint and only a few additional inquired symptoms to make disease predictions. The large sparse action problem that the flat-RL model encounters make it challenging to learn helpful candidate symptoms to inquire users, resulting in poor disease-prediction accuracy.

To solve the problems above, we also choose the hierarchical reinforcement learning based method to build an efficient and high-accuracy system. As Figure 4.2 shows, the whole system includes four main components: A master-layer agent, a group of low-layer agents, a user simulator, and a dialogue manager. As defined in Section 2.3.3, the master layer and lower layer agents work together as a hierarchy for interacting with users. More specifically, lower layer agents contain two types of models: a group of agents in the lower layer like different experts who diagnose a patient together for inquiring symptoms and making a diagnosis. Another is a classifier model for predicting diseases. Furthermore, a master agent in the master layer appoints doctors from the low layer in turn. The user simulator in the system works as the environment to directly interact with the lower layer agent. A dialogue manager manages all dialogue sessions, which mainly includes the functions of evaluation calculation and dialogue recording. Besides dynamic symptoms questions inquired by lower layer agents, the NLICE questions are designed as fixed in state representation. In addition, state representation is responsible for transferring states between different action spaces in agents.

4.3.1 Master

We defines the action space of Master as $A_m = \{w^i | i = 1,2,....,h\} \cup \{classifier\}$. h is several medical specialties we used in dataset modeling. Moreover, at each timestamp, when choosing agents that belong to medical specialty, it means that the master agent chooses to perform symptoms inquiring, whereas disease prediction when chooses "classifier" action. However, as the similar method proposed in [13], this procedure is not a Markov Decision Process as the extrinsic rewards returned during the interaction between the user and the chosen lower-layer agent can be accumulated as the immediate rewards for the master. Therefore, the reward for the master agent could be formulated as follows:

$$r_t^w = \begin{cases} \sum_{t'=1}^{N} \gamma^{t'} r_{t+t'}^e, & a_t^m = w_i, \\ r_t^e, \quad a_t^m = classifier. \end{cases} \tag{4.2}$$

where $i = 1,...,h, r_e^t$ is the extrinsic reward returned by the user at turn t, γ is the discounted factor of the master and N is the number of primitive actions of lower-layer agents.

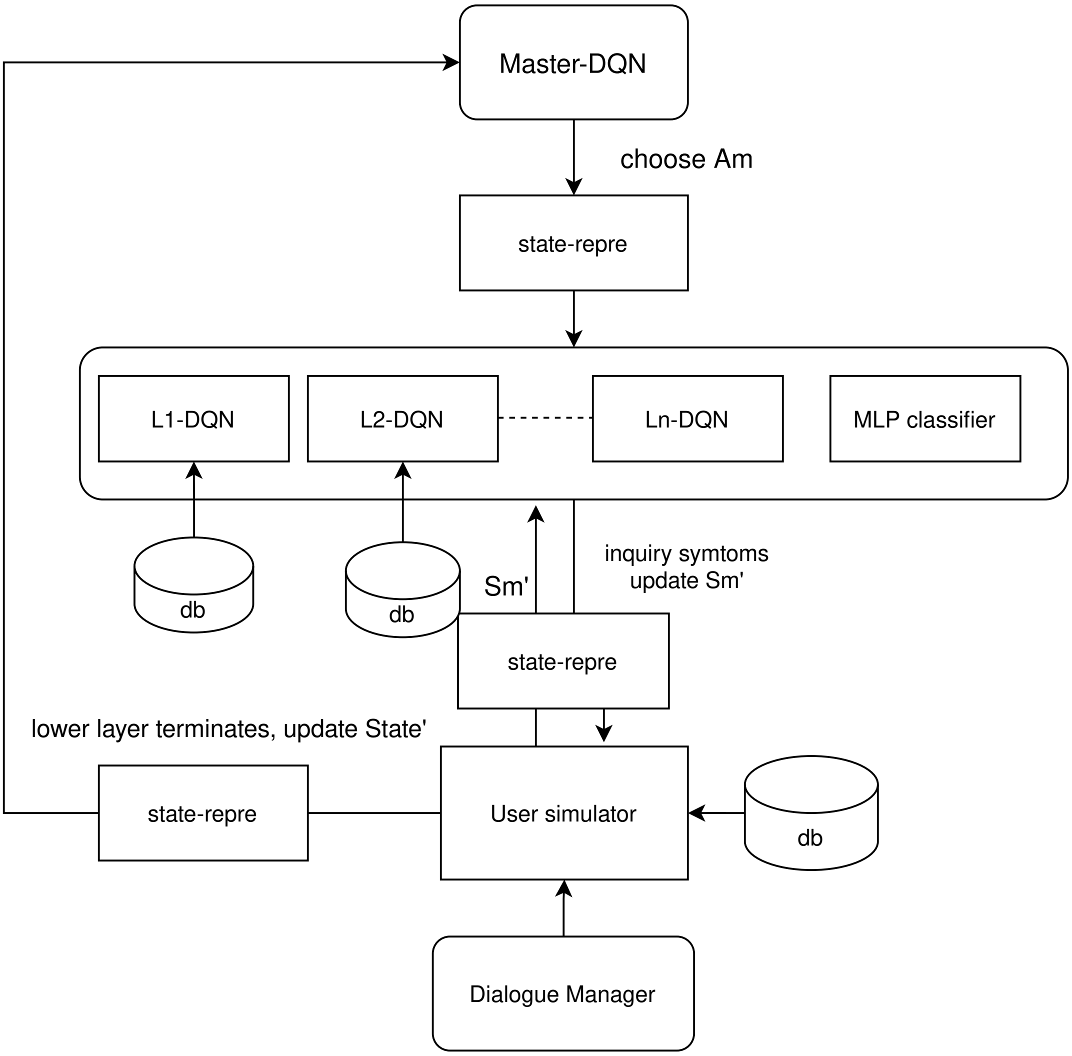

Figure 4.2: HRL system architecture

We use the same Bellman equation as defined by Equation 2.12. and loss function defined by Equation 2.13.

4.3.2 Medical specialty in lower-layer agents

There are ten types of medical specialty available in the database, as the Table 3.3 shows, each department contains numbers of diseases. The lower-layer agents w_i represents one specific medical specialty, and corresponds to the set of diseases D_i and the set of symptoms S_i. The action space of lower layer agent w_i is: $A_w^i = \{request(symptom)|symptom \in S_i\}$. At turn t, if the medical specialty w_i is activated, the current state of master s_t will be passed to that medical specialty w_i, then lower-layer agent w_i will take s_t to choose an action $a_i^t \in A_w^i$, the dialogue is updated into s_{t+1} and lower layer agent w_i will receive an intrinsic reward r_i^t from the definition of internal critic.

4.3.3 Extra disease classifier in lower-layer agents

Once the master picks the disease classifier as current action, it will take current user state s_t as input and output a vector $p \in R|D|$, representing the probability distribution among all diseases. The disease with the highest probability will inform the user of the diagnosis prediction result. We implemented two layers of Multi-Layered Perceptron (MLP) for the disease diagnosis classification.

4.3.4 User simulator

We use a user simulator as the environment to directly interact with the agent. At the beginning of each interaction episode, the user simulator randomly samples a patient record from the training samples. We randomly choose one symptom as main complaint from a user for initializing an interaction session. The simulator interacts with the agent during the dialogue based on the patient's true disease. One interaction session will be terminated as successful if the agent makes the correct prediction disease. If the informed disease is incorrect or the dialogue turn reaches the maximal turn T, it will be terminated as failed status.

4.3.5 Internal critic

The internal critic is responsible for generating intrinsic reward r_i^t to lower layer agent w_i after an action a_i^t is taken at turn t. r_i^t is positive if the lower layer agent asks a symptom that the user indeed suffers. Otherwise, if there are repeated actions generated by lower layer agent w_i or the number of subtasks turns reaches T_{sub}, r_i^t would be negative. For all other statuses, r_i^t would be zero. The internal critic also works to decide the termination condition of the lower layer agents. In our case, a lower layer agent is terminated as failed when there is repeated action produced or the number of subtask turns reaches T_{sub}. While a lower layer agent is terminated as successful when the user responds true to the symptom requested by the agent. The current lower layer agent finishes the subtask by collecting enough symptom information.

4.3.6 State representation

When inquiry correct symptom by one medical specialty, state representation will extract this patient's NLICE of this symptom and then update the latest state. Different from symptoms questions, the NLICE questions could be fixed as given a symptom, we could exactly know which NLICE question should be asked to the patient. Therefore performing the NLICE inquires is just a static process in case gathering a positive symptom.

4.4 Exploring positive symptoms with reward shaping

In reality, the number of symptoms a patient suffers from is much less than the size of symptom set S, resulting in a sparse action space. In other words, it is hard for the agent to discover the symptoms that the user indeed suffers from. To encourage master to choose a lower layer agent that can discover more positive symptoms, we follow [13] and use the reward shaping method to add supplemental reward to the original extrinsic reward while keeping the optimal reinforcement learning policy unchanged. The supplemental reward function from state st to s_{t+1} is defined as $f(s_t, s_{t+1}) = \gamma\theta(s_{t+1})\theta(s_t)$, $\theta(s)$ is the potential function and can be defined as

$$\theta(s) = \begin{cases} \gamma \times |j : b_j = [1, NLICE]|, if s \in s \cup s_t \\ \hfill 0, otherwise. \end{cases} \tag{4.3}$$

Where $\theta(s)$ counts the number of right requested symptoms for a given state s, $\gamma > 0$ is a hyper-parameter that decides the magnitude of reward shaping and S_t is the terminal state set. Therefore, the final reward function for master will be changed into $R_\theta^t = r_t + f(s_t, s_{t+1})$.

4.5 Context-aware policy transformation

However, only using a hierarchical reinforcement learning strategy in our dialogue system could not reflect the actual diagnosis procedure. We only rely on comprehension of symptom-disease relations without considering the context of medical knowledge. The symptoms that dialogue system inquiries should be related to underlying disease and consistent with medical knowledge.

4.5.1 Context modeling

When writing this thesis, there are only gender and age relationships completed in the current NLICE database. Therefore we only model context information with gender and age information. The original state representation is $s = [b^T]^T$, which only contains the status of symptoms information. In order to encode contextual information provided by patient, we need to slightly modify the state to $s = [b^T, C^T]^T$,which share similar concepts in [9]. Then given a state $s = [b^T, C^T]^T$, our master model should output Q-values for each lower-layer agent, followed by inquiring or disease prediction actions taken by lower-layer agents.

More specifically, the new part vector c comprises age,$gender$ contextual information of a patient, which is denoted as $c = [age^T, gender^T]^T$. We encode age as an integer number and gender binary(0/1). In this way, the agent could take actions based on medical knowledge instead of only relying on symptom-diseases relationships. Given the new state representation, our DQN algorithm to be changed now: for the master agent, each action a has two types one is inquiry action ($a \in A_m$) or a diagnosis action ($a = classifier$). Suppose the maximum Q-value of the outputs belongs to a medical specialty from low-layer groups. In that case, our lower-layer model asks for a series of symptoms, then obtains reward feedback from the patient, and continues to the next step. The feedback is incorporated into the next state $s_{t+1} = [b_{t+1}^T, c_T]$ according to symptom status encoding scheme. Otherwise, the maximum Q-value is identified as a diagnosis action.

4.5.2 Policy transformation strategy

Besides directly considering contextual information, the strategy for utilizing medical knowledge is implemented. Given an optimal policy π, we transform it directly to an optimal policy π^c with contextual information.

$$\pi_c^* = argmaxQ^*(s,a)p(c|a) \tag{4.4}$$

We can see that if the master action is a diagnosis action ($a = classifier$), the optimal policy π^c(considering context) can be obtained from the optimal value function Q(without considering context) by using the posterior probability distribution $p(c|d)$. In our cases, age and gender are available therefore we finally gain the policy transformation strategy as follows:

$$\pi_c^* = argmaxQ^*(s,a)p(age|a)p(gender|a) \tag{4.5}$$

Chapter 5

Experimental results and discussion

In this chapter, we discuss the experimental results obtained using the reinforcement learning models discussed previously.

5.1 Implementation details

The dataset we generated is constructed based on a symptom-disease database collected by medical experts and utilizes the patient records simulator Synthea to simulate the patient's records. According to Medical Specialty, there are 55 diseases in the database, and we classify them into ten departments. Given a patient record and all symptoms that this patient suffers, we randomly chose one of the symptoms as the main complaint and all other symptoms as implicit symptoms. A generated record for this dataset can be seen in Table 5.1. In total, the simulated dataset we constructed contains 550,000 patient records, of which 80 % for training reinforcement learning agents and 20% for evaluation models. Moreover, each kind of disease contains 10,000 records.

medical specialty	# of records	# of main complaints	# of implicit symptoms
Pulmonary	100K	1	2.33
Gastrointestinal	90K	1	2.54
Neurological	40K	1	3.42
Neuro-orthopaedic	60K	1	2.97
Rheumatological	50K	1	2.21
Urological	30K	1	2.97
Oropharyngeal	30K	1	3.55
STDs	70K	1	2.17
Oncological	40K	1	2.53
Deficiencies	40K	1	1.79
Total	550k		

Table 5.1: Overview of the NLICE Dataset. Each patient contains only 1 main complaint. # of records is the number of patients records included in this medical specialty.

We set ε for master and all the low-layer agents to 0.1 for balancing exploration and exploitation. The maximal dialogue length is set to 22, and the maximal subturn for lower-layer agents is set to 5, which means that low-layer departments maximal inquiry five different symptoms when this department is selected. As for reward configuration, the master agent will receive an extrinsic reward of +1 if informing the right predicted disease. Otherwise, it will receive -1 as extrinsic when it reaches maximal turns, or informs the user is wrong disease. In all other situations(like inquiring about wrong or right symptoms), the extrinsic reward is 0.

Furthermore, the hyper-parameter λ in reward shaping is +1. The sum of the extrinsic rewards (after reward shaping) over one subtask taken by a worker will be the reward for the master.

For the neural network architecture configuration of the master and all the lower-layers agents, We choose DQN, a three-layer network with two dropout layers, and the size of the hidden layer is 512. We choose 0.95 as The discounted factor γ for all agents in two-layer policy learning, and the learning rate is 0.0005. Additionally, all the lower-layer agents of medical specialists are trained every ten simulated epochs during the training process of the master.

Last but not least, with regards to the disease classifier, we choose a two-layer MLP network with a dropout layer the size of the hidden layer is also 512, and the learning rate is set to 0.0005. It has trained every ten epochs during the training process of the master.

5.2 Models for comparison

For evaluating and comparing the performance of various policy learning systems, we choose the following models and perform the differential diseases task.

As Table 5.2 shows, there are two machine learning models included as baseline models: Random Forest and Naive Bayes. Both machine learning models treat an automatic diagnosis task as a multi-label classification task. Moreover, the machine learning models concentrate on all symptoms that the user suffers and predicts diseases based on all symptoms. The reason for implementing machine learning based methods is to illustrate the upbound of reinforcement learning agents.

As for reinforcement learning models, we implement flat-RL agents and hierarchical RL agents with different optimization methods. Encoding state with/without NLICE features for RL methods could investigate the impact of improved symptom modeling. In addition, context-aware policy transformation experiments could take advantage of medical knowledge and historical diagnostic cases.

5.3 Overall performance

Table 5.2 illustrates the performance of all chosen models. Random forest and Naive Bayes with NLICE symptom modeling could achieve the highest Top 5 accuracy, which means that machine learning models could predict correct diseases once it obtains all explicit symptoms

Model	Context policy transformation	NLICE	Top 1	Top 3	Top 5	Reward	Turn	Matching rate
Random forest	N/A	True	74.7%	86.9%	99.8%	N/A	N/A	N/A
Naive Bayes	N/A	True	78.0%	92.4%	99.9%	N/A	N/A	N/A
flat-RL	False	False	23.1%	28.7%	33.2%	-1.69	1.46	10.7%
flat-RL	False	True	28.2%	35.4%	36.9%	-0.75	1.98	18.9%
flat-RL	True	True	36.2%	41.4%	45.3%	-0.74	2.34	20.1%
HRL	False	False	45.4%	52.3%	61.3%	0.20	10.69	77.1%
HRL	False	True	71.8%	87.0%	88.6%	0.52	9.95	84.6%
HRL	True	True	83.9%	90.5%	93.8%	0.87	9.89	88.7%

Table 5.2: Overall performance

and the main complaint. Figure 5.1 represents performance of RL Models with proposed NLICE modeling approach and context-aware policy.

As Figure 5.1 shows, flat-RL performs negatively with only around 23.1 % Top 1 success rate. NLICE symptom modeling could only improve the performance slightly to 28.2%. Remarkably, for all flat-RL experiments, the average steps are less than 3, which means that flat-RL agent tends to directly predict diseases based on the primary complaint without asking for additional explicit symptoms. We hypothesize that the flat-RL model encounters difficulties learning additional positive symptoms to ask users due to the sparse space of symptoms and action, resulting in poor disease-prediction accuracy.

Compared to the flat-RL models, HRL model takes more steps to have longer interactions with the user so that it can obtain more information about their implicit symptoms.With more information about implicit symptoms, our HRL the model significantly outperforms the other baselines in the Top 1 diagnosis success rate. There is also an increase in the matching rate as the number of steps rises. Furthermore, the HRL model reaches 71.8% Top 1 success rate after encoding NLICE symptoms, proving the efficiency of introducing NLICE symptoms and HRL-based methods.

We also report the results of context-aware policy transformation via modeling context and strategy mentioned in Chapter 4.5. Compared with the task without a context-aware policy transformation strategy, we can see that the accuracy results for flat-RL and HRL are improved by at least five percentage points with context-aware policy transformation.

5.4 Further analysis

5.4.1 NLICE modeling case study and performance analysis

To compare the performance of the RL based models, we choose multiple patient records in the test set of our generated dataset and output the dialogue interaction procedures. Table 5.4.1 represents a failed diagnosis case via using flat-RL method with NLICE symptom modeling. We could observe that flat-RL agent only inquires about one additional symptom, "Altered breathing," then directly predicts disease. Unfortunately, it predicts wrongly

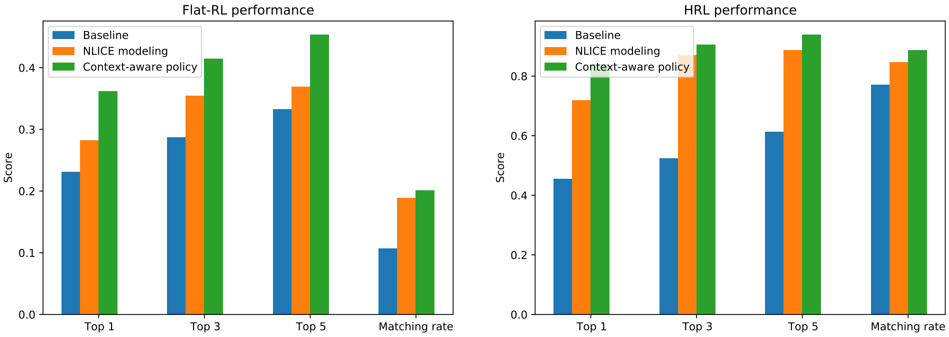

Figure 5.1: Performance of RL Models with proposed NLICE modeling approach and context-aware policy

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Cough	N/A	main complaint
1	Altered breathing	”nature”:”Dyspnea”	Yes
Top 5	Disease		
1	**Pneumonia (Pneumococcal most likely)**		
2	Pneumonia (Adenovirus most likely)		
3	Asthma		
4	COPD		
5	Lung cancer (NSCLC)		

Table 5.3: A failed interaction example using the flat-RL method. The true disease is Asthma and the main complaint is Cough.

in top-1. Patients who suffer from Pneumonia (Pneumococcal most likely) and Pneumonia (Adenovirus most likely) usually also have Cough and Altered breathing. The nature of Altered breathing is the same as Asthma. Therefore, even though the NLICE modeling approach is implemented in flat-RL model, it is still difficult for flat-RL to distinguish those diseases within a limited number of inquiring symptoms.

Table 5.4 represents the HRL interaction procedures. The true disease of this patient is Wernicke-Korsakoff complex, and the main complaint is Paresis Paralysis. The number of additional symptoms inquired by the agent is far more than the flat-RL agent, therefore predicting the disease correctly.

We then compare the impact of NLICE symptom modeling by using the same test case. This patient’s true disease is Pneumonia (Adenovirus most likely), and the main complaint is Fever. The Table 5.5 and Table 5.6 represent HRL agent with NLICE symptom modeling, without NLICE symptom modeling respectively. We could find that HRL agent usually lacks the ability to distinguish the diseases that share similar symptoms. One example is

Inquiry Turn	Inquiry Symptom	NLICE	Response
0	Paresis Paralysis	”location_main”: ”Head” ”location”: ”Eyes”	main complaint
1	Altered movement	”nature”: ”Ataxia”, ”location_main”: ”Head”, ”location”: ”Eyes”	Yes
2	Fever	N/A	False
3	Self diagnosis	N/A	False
4	Altered breathing	N/A	False
5	Swelling	N/A	False
6	Altered mental state	”nature”: ”Disorientation”	Yes
Top 5	Disease		
1	**Wernicke-Korsakoff complex**		
2	Subarachnoidal bleeding		
3	Pulmonary edema (among oldest)		
4	Otitis media acuta		
5	viral respiratory infection (Influenza most likely)		

Table 5.4: A successful interaction example. The true disease is Wernicke-Korsakoff complex and the main complaint is Paresis Paralysis.

Pneumonia (Adenovirus most likely), the wrongly informed diseases in Table 5.6 all contain the same symptoms as Pneumonia (Adenovirus most likely). In contrast, the agent successfully predicts correct disease after NLICE encoding in Table 5.5 because those similar symptoms contain different NLICE features.

Besides the case study, we also report the learning trends of NLICE symptom modeling experiments in Figure 5.2. There is a noticeable increase in NLICE modeling approach compared with no NLICE modeling, which indicates that our NLICE symptom modeling approach is beneficial to the diagnosis accuracy.

5.4.2 Performance of lower-layer agents

In order to analyze deeply the patient who has been informed of the wrong disease by the master agent in the hierarchical reinforcement learning scheme, we collect all the wrong predicted user diseases. It illustrated the error matrix in Figure 5.3, 5.4 , 5.5, 5.6. Four tasks are formed as a benchmark.

It shows the disease prediction result for all the ten medical specialty groups. We can see the color of the diagonal square is darker in tasks without NLICE symptom modeling, especially in Pulmonary and Neurological. The diseases within these two groups often share similar and even the same symptoms. Therefore not easy to distinguish them when NLICE symptoms are not included. Context-aware polity transformation could partly eliminate the miss classification problem for highly sex-age-related diseases. Nevertheless, the color of

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Fever	N/A	main complaint
1	Cough	”nature”: ”Mucus”,	Yes
2	Congestion	N/A	False
3	Pain	”nature”: ”Myalgia”, ”location_main”: ”Head”	Yes
4	Altered breathing	N/A	False
5	Swelling	N/A	False
Top 5	Disease		
1	**Pneumonia (Adenovirus most likely)**		
2	Pneumothorax		
3	viral respiratory infection (Influenza most likely)		
4	Viral meningitis (Varicella zoster virus)		
5	Lumbago (Muscle strain)		

Table 5.5: A successful interaction example. The true disease is Pneumonia (Adenovirus most likely) and the main complaint is Fever.

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Fever	N/A	main complaint
1	Pain	N/A	Yes
2	Nausea	N/A	False
3	Swelling	N/A	False
4	Altered breathing	N/A	False
5	Fatigue	N/A	False
Top 5	Disease		
1	**viral respiratory infection (Influenza most likely)**		
2	viral respiratory infection (COVID-19 most likely		
3	Testicular cancer		
4	Viral meningitis (Varicella zoster virus)		
5	Migraine		

Table 5.6: A failed interaction example without NLICE modeling. The true disease is Pneumonia (Adenovirus most likely) and the main complaint is Fever.

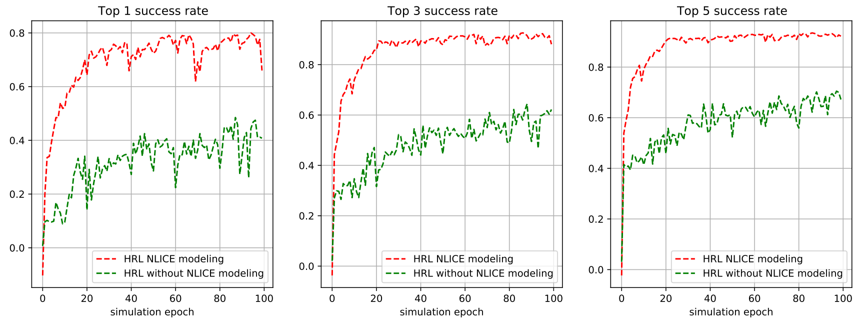

Figure 5.2: Learning curves of HRL models

the wrong classification in Pulmonary medical specialty is still the darkest among all other groups after NLICE modeling and context-aware policy transformation because the diseases in Pulmonary group usually contain the same symptom even with NLICE modeling.

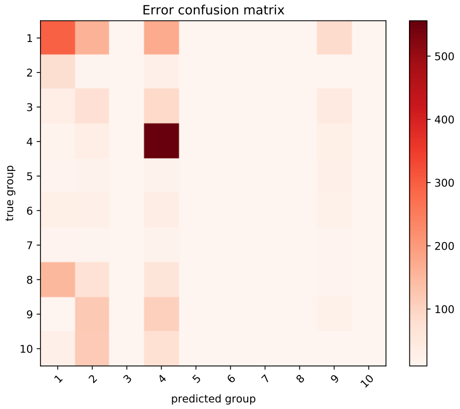

Figure 5.3: without NLICE modeling task

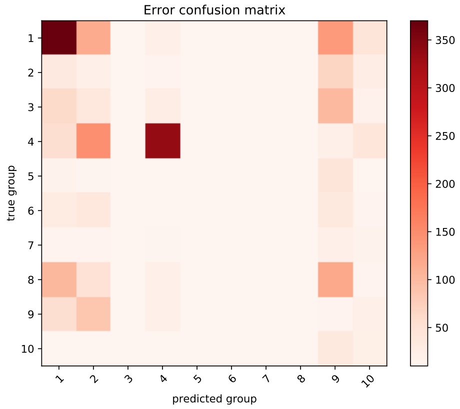

Figure 5.4: without NLICE modeling task after context-aware policy transformation

5.4.3 Case study for policy transformation

The context information includes age and gender based on the currently available dataset. Some examples below illustrate how context-aware policy transformation could change the diagnosis prediction.

Table 5.7 and Table 5.8 uses a same patient who suffers disease which is highly related to age factor. We could see that the true disease is Pulmonary edema (among old), and this patient is 69 years old. As medical experts defined in the original data set, Pulmonary edema (among old) could only occur in patients from 65 to 84. However, without con-

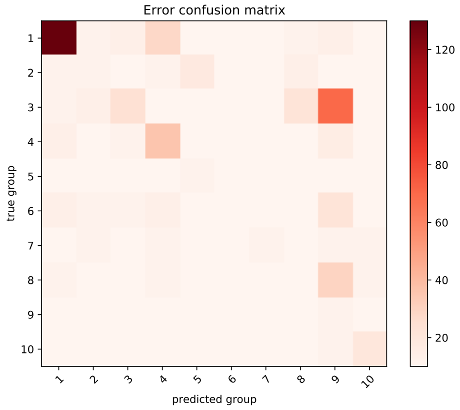

Figure 5.5: NLICE modeling

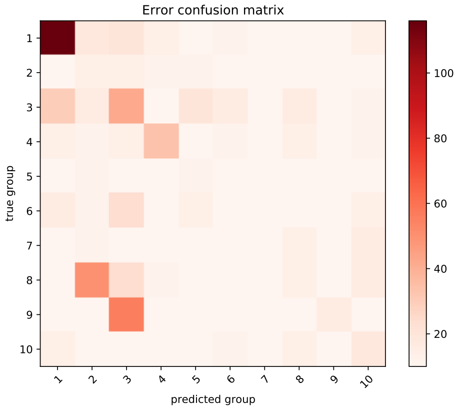

Figure 5.6: NLICE modeling task after context-aware policy transformation

textual information, the agent wrongly predicts Top 1 disease as Pulmonary edema (among young) that only occurs in patients below 64. When we consider the context background in Table 5.7, the target disease Pulmonary edema (among old) is informed successfully as Top 1.

The above examples represent cases that are highly influenced by age distribution. In Table 5.9 and Table 5.10 we show another case in which context policy transformation fixes the wrongly predicted diagnosis result. If the predictions do not consider gender context, the Top 5 predictions wrongly inform the disease targeting the wrong gender. Conversely, Rheumatoid arthritis (M) is moved to the Top 1 result when the context is considered since the patient is male.

However, context-aware policy transformation sometimes also misleads the diagnosis by suppressing the probability of diseases highly correlated with age and gender. For example, Table 5.8 wrongly moves the ”Testicular cancer” to Top 2, and the gender of this patient is male. Furthermore, Testicular cancer also only appears in males. However, apparently, the symptoms of Testicular cancer do not include Chest Pain, Dyspnea Altered breathing.

5.4.4 Performance of reward shaping

To evaluate the learning efficiency and accuracy of the reward shaping algorithm, we form two diagnostic tasks: one uses the reward shaping algorithm during the interaction loop. In contrast, another one uses a basic reward configuration as defined in 4.3.1. Figure 5.7 illustrates the learning curves of these two tasks. The experimental result shows that using the reward shaping algorithm task outperforms the baseline. Table 5.11 shows the comparison of cumulative discounted rewards received in the interaction loop. We can see that the HRL baseline struggles to gain positive rewards. However, our reward shaping algorithm can obtain rewards far more than the basic reward configuration. This proves that the reward

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Pain	”location main”: chest	main complaint
1	Altered breathing	”nature”:”Dyspnea”	Yes
2	Cough	N/A	False
3	Dizziness	N/A	Yes
4	Eruption	N/A	False
5	Fever	N/A	False
Top 5	Disease		
1	**Pulmonary edema (among young)**		
2	Pulmonary edema (among old)		
3	Pulmonary edema (among oldest)		
4	viral respiratory infection (COVID-19 most likely)		
5	Vitamin B12 deficiency		

Table 5.7: A failed interaction example without context-aware policy transformation. The true disease is Pulmonary edema (among old) and the main complaint is Pain. The patient is 69 years old.

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Pain	”location main”: chest	main complaint
1	Altered breathing	”nature”:”Dyspnea”	Yes
2	Cough	N/A	False
3	Dizziness	N/A	Yes
4	Eruption	N/A	False
5	Fever	N/A	False
Top 5	Disease		
1	**Pulmonary edema (among old)**		
2	Testicular cancer		
3	Female cystitis		
4	Pulmonary edema (among oldest)		
5	Carpal tunnel syndrome		

Table 5.8: A successful interaction example with context-aware policy transformation. The true disease is Pulmonary edema (among old) and the main complaint is Pain. The patient is 69 years old.

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Altered_appetite	”nature”:”Decreased”	main complaint
1	Pain	”location_main”:”Joints”	Yes
2	Altered_breathing	N/A	False
3	Swelling	N/A	Yes
4	Nausea	N/A	False
5	Cough	N/A	False
6	Dizziness	N/A	False
7	Fever	N/A	False
Top5	Disease		
1	**Rheumatoid arthritis (F)**		
2	Rheumatoid arthritis (M)		
3	Otitis media acuta		
4	Oesophageal carcinoma		
5	Vitamin D deficiency		

Table 5.9: A failed interaction example without context-aware policy transformation. The true disease is Rheumatoid arthritis (M) and the main complaint is Pain. The gender of the patient is male.

Inquiry			
Turn	Inquiry Symptom	NLICE	Response
0	Altered_appetite	”nature”:”Decreased”	main complaint
1	Pain	”location_main”:”Joints”	Yes
2	Altered_breathing	N/A	False
3	Stiffness	”location main”:”Joints” ”exciation”:”Morning”	Yes
4	Nausea	N/A	False
5	Cough	N/A	False
6	Dizziness	N/A	False
7	Fever	N/A	False
Top5	Disease		
1	**Rheumatoid arthritis (M)**		
2	Rheumatoid arthritis (F)		
3	Oesophageal carcinoma		
4	Vitamin D deficiency		
5	Otitis media acuta		

Table 5.10: A successful interaction example with context-aware policy transformation. The true disease is Rheumatoid arthritis (M) and the main complaint is Pain. The gender of the patient is male.

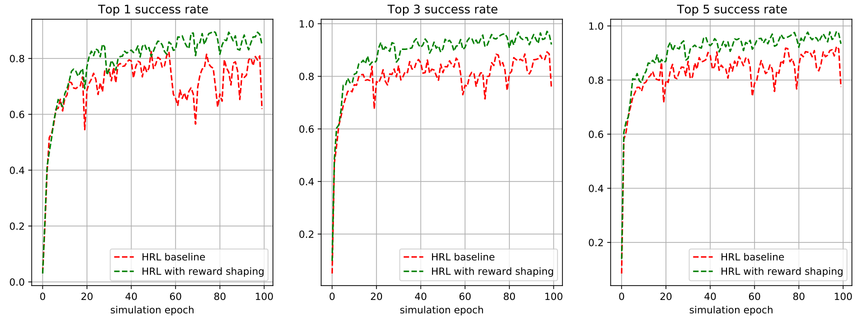

Figure 5.7: Learning curves of HRL models for studying reward shaping performance

model	task	average cumulative discounted rewards
HRL	basic reward configuration	-3.65
	reward shaping	0.87

Table 5.11: Average cumulative discounted rewards of baseline and reward shaping task

shaping algorithm plays a positive role in discovering actual symptoms and could partly solve the sparse action space problem in the diagnosis task.

5.4.5 Performance of varying different number of main complaints

Recall that for dataset statistics in Table 5.1, the number of the main complaint is set to 1 by default. The average number of implicit symptoms for each medical specialty varied from 2 to 5. Even though the hierarchical reinforcement learning scheme with multiple optimization approaches could efficiently improve the performance, the experiments are still limited to one simulated scenario because only one main complaint is chosen. Generally, it is not possible that all patients went to the hospital with only one known symptom.

To investigate the model performance under various scenarios in the real world, we change the number of main complaints by adjusting the user simulator configuration. Instead of randomly choosing one symptom from all symptoms as the main complaint, we create two more dataset settings where two and three symptoms are chosen as main complaints, and all other true symptoms are set as implicit ones. All other experimental setting is as the same as the setting in Chapter 5.1.

We report the learning curves of three data settings in Figure 5.8. MLP performance and Average Turn could be found in Figure 5.9. As we expected, the performance of Top n success rate and MLP accuracy increases slightly as the number of main complaints grows. However, all scenarios could finally achieve over 90 % Top 5 success rate. It is worth mentioning that the average turn decreases significantly when the number of main complaints increases. We conclude that if more symptoms are known when starting the diagnosis procedures, the agent tends to inquire shorter sequence of symptoms and then make a diagnosis

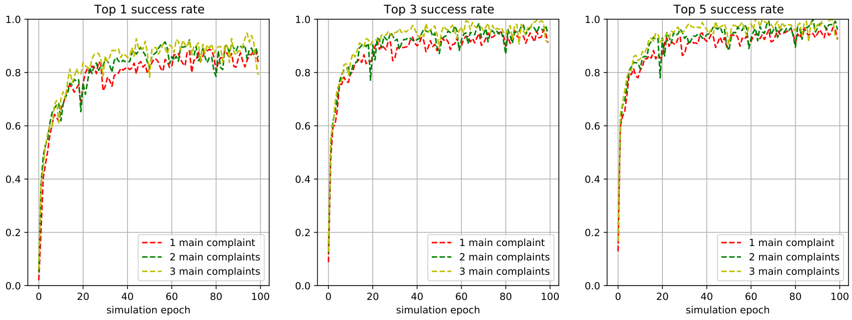

Figure 5.8: Learning curves of HRL models with varying different number of main complaints

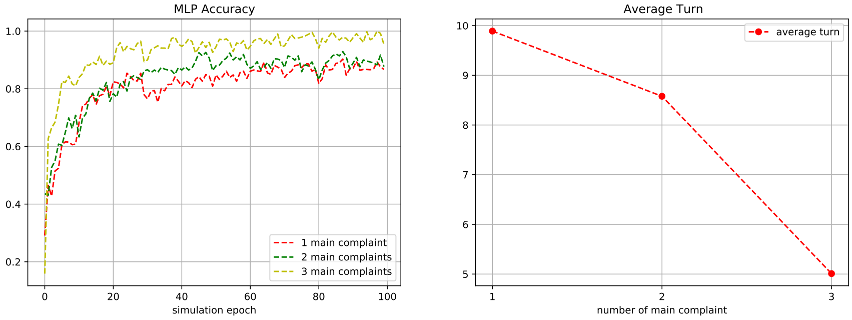

Figure 5.9: MLP learning curve and average turn with varying different number of main complaints

decision. Nevertheless, the number of complaints could not significantly increase the Top n success rate because the hierarchical RL approach could efficiently find positive symptoms with longer interaction turns.

5.4.6 Effect of increased maximum number of turn

HRL models could decide the maximum number of inquiries during the interaction process. We vary the maximum number of turn(5,10,15,20,25) in the HRL configuration setting, and compare the Top n results of increased maximum number of turn in Figure 5.10, Average reward that master agent gained in Figure 5.11 and matching rate results in Figure 5.12. The baseline model is the HRL model with NLICE symptom modeling. The reward shaping task adds a reward shaping algorithm to the baseline model, and the context-aware policy task employs a context-aware policy strategy the reward shaping task. Similar to previous experimental results, there is a gradual rise from the baseline model to the context-aware policy task. Top n success rates increase on all three tasks with larger interaction turns, and finally could achieve above 80% Top 1 in the maximum 25 turns. However, the average

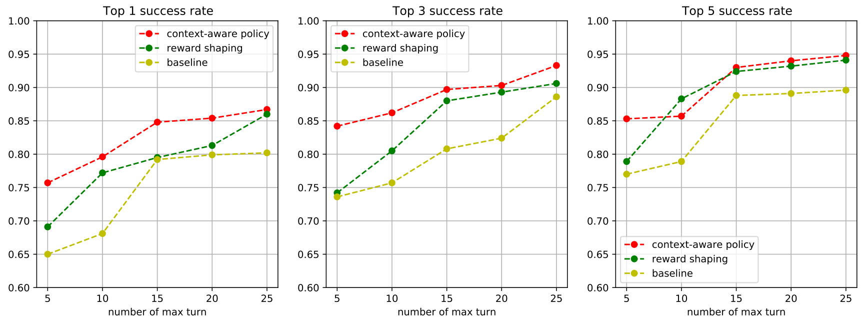

Figure 5.10: Performance of varying different number of max turn

reward of the baseline task decreases when increasing the number of turns, which means that the agent intends to ask more wrong symptoms, thereby leading negative reward. At the same time, the matching rate and Top n success rate remain almost unchanged in the baseline model after 15 turns. In addition, for the other two tasks, average reward results first rise as the increasing turns peak at 15 turns. However, the average reward decreases gradually with more interactions.

Our HRL models could gain enough positive symptoms within 15 rounds. Even though Longer question sequences could slightly increase the Top n success rate and matching rate, patients are sometimes impatient to answer long questionnaires. We conclude that our system could provide a user-friendly symptom inquiring process and maintain a high accuracy-precision diagnosis prediction.

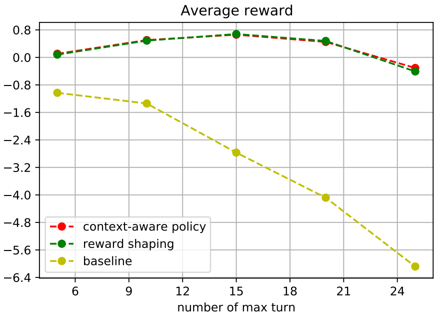

Figure 5.11: Average reward results of varying different number of max turn

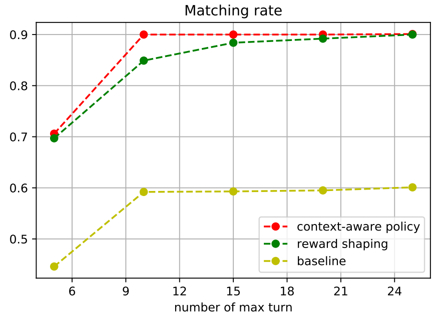

Figure 5.12: Matching rate results of varying different number of max turn

5.5 Conclusions

We report the overall performance in Figure 5.13,5.14. As the figures show, the proposed HRL model that enables different experts jointly diagnose significantly improves diag-

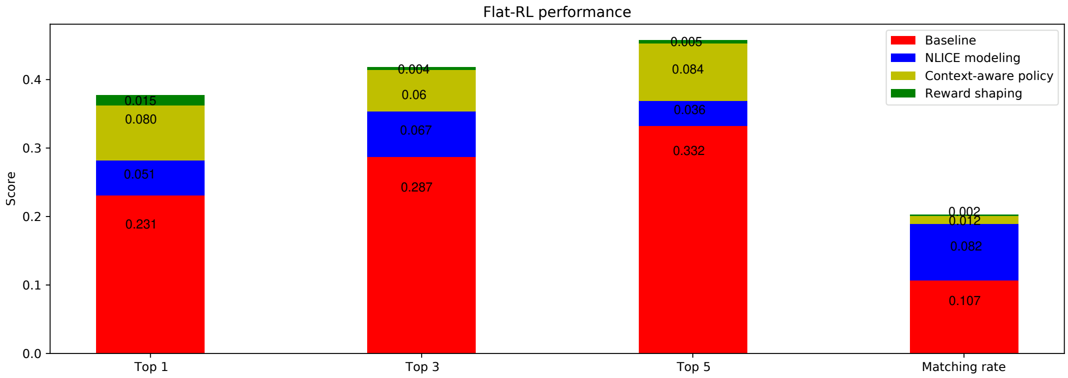

Figure 5.13: Overall performance in flat-RL Model with proposed improvements

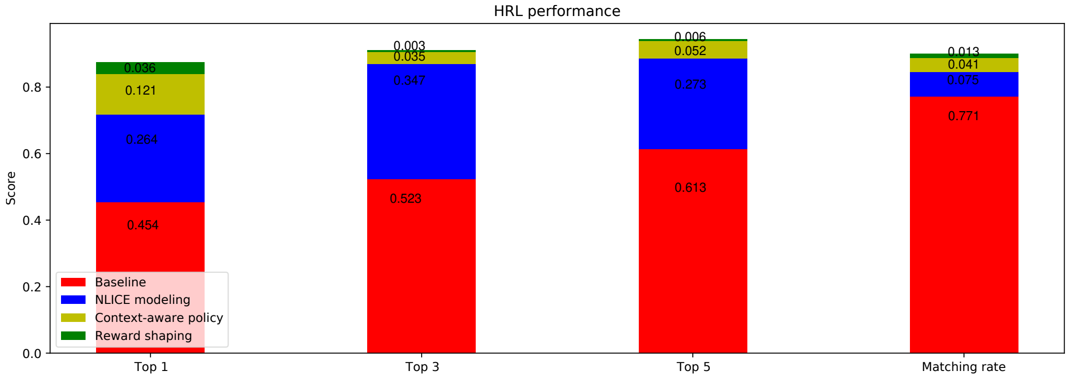

Figure 5.14: Overall performance in HRL Model with proposed improvements

nostic accuracy compared to the flat-RL model while tending to generate shorter inquiry sequences. Obviously, the HRL model outperforms the flat-RL model in matching rate, which means that the HRL model can explore more positive symptoms compared with the flat-RL model. Furthermore, the NLICE symptom modeling method is beneficial for identifying diseases that share similar symptoms, as few diseases completely contain the same symptoms after NLICE modeling. More specifically, flat-RL and HRL models increase Top 1 success rate by 5% and 26.2%, respectively. When considering contextual information(gender and age), we have shown transfer the optimal policy to utilize the contextual information plays a positive role in improving performance. However, the trade-off between wrongly dominating age-sex highly related diseases and symptom dominating optimal policy still needs to be investigated in the future. The reward shaping algorithm is more suitable for tasks with more interaction loops. Therefore flat-RL shows limited improvement with the reward shaping algorithm, whereas HRL increases around 5% Top 1 success rate. The same trends of using optimization approaches could be found in Top 3 success rate and Top 5 success rate.

Chapter 6

Conclusions and Future Work

This chapter gives an overview of the thesis conclusions and contributions. Then, some ideas for future work will be discussed.

6.1 Conclusions

In this thesis, we firstly formulate differential diagnosis as a Markov Decision Process and implement flat and hierarchical policy learning approaches to perform interactions between patient and agent. The flat-RL approach treats all symptoms and diseases as action, which is easily implemented but results in large space action problems. Therefore, we further employ hierarchical reinforcement learning algorithms to split a complicated task into several subtasks in a two-layer policy learning manner: symptom inquiry and predicting disease. The sub-task of disease prediction is assigned to different kinds of workers in the lower hierarchy level. The high level of the hierarchy is a master agent who takes charge of deciding which lower layer agent should directly interact with a patient.

The experimental results shown in Chapter 5 conclude that the flat-RL algorithm is only suitable for trivial diagnosis task. The flat-RL agent generates shorter inquiry sequences and fails to explore positive symptoms due to the sparse action space. In contrast, hierarchical RL could partly solve the problems involved in a flat-RL agent and improve the success rate from 23.1% to 45.4% Top 1 success rate.

Besides the algorithms, we generate a reliable patient dataset to evaluate our reinforcement learning models. Moreover, different from the available public medical dataset only based on symptom-disease probability relationships, we firstly propose a novel symptom modeling method and represent symptoms with more characters. The performance results also prove the positive effect made by NLICE symptom modeling approach: The flat-RL method slightly increases the Top 1 success rate from 23.1% to 28.2%. Remarkably, there is a significant improvement (from 45.4% to 71.8%) in HRL method after modeling NLICE symptoms. In order to power the proposed algorithms, we further introduce demographic background of patients and then utilize contextual information to transfer the optimal policy, which is beneficial to identifying the diseases that are highly related to age and gender. Finally, our HRL models could achieve more than 90% Top 5 success rate after context-aware policy transformation.

The reward shaping algorithm is an alternative way to solve the sparse action space. Therefore we choose an informative potential reward shaping algorithm to guide the HRL agent to explore the sparse action space productively. As the results show, the reward shaping algorithm increases the average reward gained from -3.65 to 0.87. In addition, the shaping algorithm increases the Top n success rates by around 10%.

We discuss the trade-off between a user-friendly system and high-precision prediction by varying the maximum number of interaction loops. The results show that the longer sequence can contribute to higher accuracy when gaining more symptom information but reduce the average reward. Moreover, our system can gain enough positive symptoms within 15 rounds of questions, proving the efficiency of our proposed optimization approaches.

Last but not least, we report the effect of the number of main complaints in Chapter 5. The main aim of this experiment is to investigate the action of models in a real-world scenario. The results indicate that our HRL model tends to inquire fewer questions when patients provide enough initial complaints, reflecting the realistic scenario.

6.2 Future work

In future work, there are multiple research directions could be explored for the task of automatic diagnosis.

Expand dataset with more diseases and symptoms

The experimental results are based on a database that consists only 55 conditions. There is still significant room for expanding the database with more diseases and symptoms to investigate the efficiency and suitability of reinforcement learning for automatic diagnosis tasks. Furthermore, improving the reward shaping algorithm to deal with the larger action space could be further explored and optimized.

From contextual information to doctor experience

Current available contextual information in the database is limited to gender and age, so disease related to gender and age will be heavily influenced after context-aware policy transformation. It might be more worth researching the contextual effect if we have more attributes in the future. The context-aware policy transformation will consider more factors instead of being heavily influenced by age and gender.

The complicated automatic medical diagnosis task needs more critical requirements for the dialogue rationality in the context of medical knowledge and symptom-disease relations. Therefore, the need for doctor experience is increasingly important in later research. And utilizing doctor experience and medical knowledge to form a knowledge graph might be a promising research area in the medical domain.

Reformulate automatic diagnosis as a sequence generation task

In this project, we formulate automatic diagnosis as a Markov Decision Process and implement a reinforcement learning approach to perform multi-step inquiry and diagnosis. Even though we could achieve over 80% Top 1 accuracy, limitations such as low efficiency, repeated actions, and complex reward shaping configuration still occur in the system.

As an automatic diagnosis system is built on conversations between the agent and the patient, we could formulate this problem in a different task: Symptom Generation task, which takes advantage of natural learning processing techniques to address the problems mentioned before. The objective of the sequence generation task is to generate a target sequence given a source input, which is suitable for automatic diagnosis background.

Bibliography

[1] Obinna Agba. Effective primary healthcare differential diagnosis: A machine learning approach. 2020.

[2] Kathi Canese and Sarah Weis. Pubmed: the bibliographic database. *The NCBI handbook*, 2(1), 2013.

[3] Sudesna Chatterjee, Kamlesh Khunti, and Melanie J Davies. Type 2 diabetes. *The lancet*, 389(10085):2239–2251, 2017.

[4] Fabrizia Faustinella and Robin Jacobs. The decline of clinical skills: a challenge for medical schools. *International Journal of Medical Education*, 9:195–197, 07 2018. doi: 10.5116/ijme.5b3f.9fb3.

[5] Robert Ferrari et al. Effect of a symptom diary on symptom frequency and intensity in healthy subjects. *The Journal of rheumatology*, 37(11):2387–2389, 2010.

[6] Barbara Herlihy. *The Human Body in Health and Illness-E-Book*. Elsevier Health Sciences, 2017.

[7] Ronald A Howard. Dynamic programming and markov processes. 1960.

[8] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3 (1):1–9, 2016.

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

[10] R Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. URL `https://www.osti.gov/biblio/421279`.

[11] Igor Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7:317–337, 1993.

[12] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001. ISSN 0933-3657. doi: https://doi.org/10.1016/S0933-3657(01)00077-X. URL `https://www.sciencedirect.com/science/article/pii/S093336570100077X`.

[13] Kangenbei Liao, Qianlong Liu, Zhongyu Wei, Baolin Peng, Qin Chen, Weijian Sun, and Xuanjing Huang. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning, 2020. URL `https://arxiv.org/abs/2004.14254`.

[14] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. EDDI: efficient dynamic discovery of high-value information with partial VAE. *CoRR*, abs/1809.11142, 2018. URL `http://arxiv.org/abs/1809.11142`.

[15] Kathleen M McTigue, Robert Wellman, Elizabeth Nauman, Jane Anau, R Yates Coley, Alberto Odor, Julie Tice, Karen J Coleman, Anita Courcoulas, Roy E Pardee, et al. Comparing the 5-year diabetes outcomes of sleeve gastrectomy and gastric bypass: the national patient-centered clinical research network (pcornet) bariatric study. *JAMA surgery*, 155(5):e200087–e200087, 2020.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[17] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, 10, 1997.

[18] Hannah Semigran, Jeffrey Linder, Courtney Gidengil, and Ateev Mehrotra. Evaluation of symptom checkers for self diagnosis and triage: Audit study. *BMJ (Clinical research ed.)*, 351:h3480, 07 2015. doi: 10.1136/bmj.h3480.

[19] Hannah Semigran, Jeffrey Linder, Courtney Gidengil, and Ateev Mehrotra. Evaluation of symptom checkers for self diagnosis and triage: Audit study. *BMJ (Clinical research ed.)*, 351:h3480, 07 2015. doi: 10.1136/bmj.h3480.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[21] Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998.

[22] Kai-Fu Tang. Inquire and diagnose : Neural symptom checking ensemble using deep reinforcement learning. 2016.

[23] Martijn van Otterlo and Marco Wiering. *Reinforcement Learning and Markov Decision Processes*, pages 3–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_1. URL https://doi.org/10.1007/978-3-642-27645-3_1.

[24] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018.

[25] Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 05 1992. doi: 10.1007/BF00992698.

[26] 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)*, pages 201–207, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2033. URL https://aclanthology.org/P18-2033.