Context-Aware Symptom Checking for Disease Diagnosis Using Hierarchical Reinforcement Learning

Hao-Cheng Kao*
HTC Research & Healthcare
haocheng_kao@htc.com

Kai-Fu Tang*
HTC Research & Healthcare
kevin_tang@htc.com

Edward Y. Chang
HTC Research & Healthcare
edward_chang@htc.com

Abstract

Online symptom checkers have been deployed by sites such as WebMD and Mayo Clinic to identify possible causes and treatments for diseases based on a patient's symptoms. Symptom checking first assesses a patient by asking a series of questions about their symptoms, then attempts to predict potential diseases. The two design goals of a symptom checker are to achieve a high accuracy and intuitive interactions. In this paper we present our context-aware hierarchical reinforcement learning scheme, which significantly improves accuracy of symptom checking over traditional systems while also making a limited number of inquiries.

Introduction

With the quantity of information available online, self-diagnosis of health related ailments has become increasingly prevalent. According to a survey in 2012 (Semigran et al. 2015), 35% of U.S. adults in the U.S. have attempted to self-diagnose their ailments through online services This process often starts by searching a particular symptom on a search engine. While online searches have fast accessibility and require no cost, search quality can potentially be dissatisfactory since search results could be irrelevant, imprecise or even incorrect.

As stated in (Ledley and Lusted 1959), there are three components in a disease diagnosis process: (i) medical knowledge, (ii) signs and symptoms presented by the patient, and (iii) the final diagnosis itself. We refer to such processes as *symptom checking*, and refer to an agent capable of performing such diagnoses as a *symptom checker*. In a symptom checker, a medical knowledge base serves as the source of medical knowledge, which depicts the probabilistic relationship between symptoms and diseases. An inference engine is responsible for formulating symptom inquiries, collecting patient information, and then performing diagnosis by utilizing both the individual's information and the medical knowledge base. If the prediction confidence is not high, the inference engine may suggest conducting relevant lab tests to facilitate diagnosis. Finally, the diagnosis process outputs a list of potential diseases that the patient may have.

*The first two authors contributed equally to this paper.

The primary goal of a symptom checker is to achieve high disease-prediction accuracy. Attaining the best possible accuracy requires full information about a patient, including not only his/her symptoms, but also his/her medical record, family medical history, and lab tests. However, an online symptom checker may only be able to obtain a list of symptoms, and therefore must rely on partial information. This lack of information often means the online symptom checker cannot attain extremely high accuracy. At the same time, even obtaining a list of symptoms in a user friendly manner is a challenge, since there are over one hundred different medical symptoms, and few patients may be willing to fill out such a long symptom questionnaire. Consequently, this problem leads to the second requirement for an effective symptom checker, good user experience. The primary consideration in achieving good user experience is for a symptom checker to make only a limited number of inquiries. The design goal is then to maximize information gain when only a limited number of symptom inquiries can be made to achieve high diagnosis accuracy.

In previous works (Kononenko 2001; Kohavi 1996; Kononenko 1993), Bayesian inference and decision trees as well as entropy or impurity functions were proposed to select disease symptoms and to perform diagnoses. However, these works generally considered only local optimums by some means of greedy or approximation schemes. These approaches often result in compromised accuracy. Expert systems are also used in medical diagnosis systems (Hayashi 1991). In this regime, rule-based representations are extracted from human knowledge or medical data. The final inference quality depends on the quality of the extracted rules. For example, in (Hayashi 1991), if-else rules are extracted from fuzzy neural networks learned from medical data. Their rule-based representation focuses on knowledge acquisition and does not pursue a shorter section of interactions with users. Our prior work (Tang et al. 2016) proposes *neural symptom checking*, adopting reinforcement learning to simultaneously conduct symptom inquiries and diagnose. The optimization objective captures a combination of inquiry length and diagnosis accuracy. Though our top-1 accuracy reaches 48% (for 73 diseases), which is higher than the 34% average accuracy achieved by online services surveyed by a Harvard report (Semigran et al. 2015), substantial room exists for further improvement. (Table 1 presents details.)

Online Services	#Diseases	Accuracy
Esagil	100	20%
MEDoctor	830	5%
Mayo Clinic	N/A	36%
WebMD	N/A	17%
Harvard Report (Avg.)	N/A	34%

Table 1: The current status of online diagnosis services evaluated by the Harvard report (Semigran et al. 2015). The top-1 accuracies of representative sites are shown in percentages. Note that Esagil and MEDoctor are the only services that explicitly disclose the number of their supported diseases. The four listed services require a full list of the patients symptoms.

In this work, we introduce two novel enhancements to improve diagnosis accuracy. First, we introduce a latent layer using anatomical parts. We employ hierarchical reinforcement learning to use a committee of anatomical parts to make a joint diagnostic decision. While each anatomical part model is capable of selecting symptoms to inquire and diagnose within the expertise of its anatomical part, each can also inquire about different symptoms in a given state. Our proposed model utilizes a master model to select the specific anatomical part on which to perform inquiries at each interaction step. Our second enhancement is to introduce *context* into the model and make our symptom checker context aware. The contextual information includes, but is not limited to, three aspects about a patient: *who*, *when*, and *where*. The who aspect includes a person's demographic information (e.g., age and gender), heredity (characterized by genetic data), and medical history. The when aspect can be characterized by a distribution of diseases in the time of year. The where aspect can be characterized by a distribution of diseases from coarse to fine location granularities (e.g., by country, city, and/or neighborhood). Any joint distributions of any combinations of the who, when and where aspects can be formulated and quantified into a context-aware model. Empirical studies on a simulated dataset show that our proposed model drastically improves disease prediction accuracy by a significant margin (for top-1 prediction, the improvement margin is 10% for 50 common diseases[1] and 5% when expanding to 100 diseases).

The rest of this paper is organized into five sections. We first briefly review formulating symptom checking in a reinforcement learning problem. We then introduce our proposed hierarchical ensemble model of anatomical parts. Next, we present the context-aware model. The experiment section presents our results. An earlier version of our symptom checker is employed in our DeepQ Tricorder (Chang et al. 2017), which was awarded second prize in the Qualcomm Tricorder XPrize Competition (Qualcomm 2017). Finally, we offer our concluding remarks.

[1]The term common disease here means frequently occurred diseases from the Centers for Disease Control and Prevention (CDC) dataset.

Reinforcement Learning Formulation

We regard a symptom checker as an agent solving a sequential decision problem. This agent interacts with a patient as follows: Initially, the agent is provided with a symptom that the patient may have from the set of all symptoms $\mathcal{I}$. This provided symptom is regarded as the initial symptom. In each time step, the agent chooses a symptom $i \in \mathcal{I}$ to inquire the patient about. The patient then responds to the agent with a true/false answer indicating whether he/she is suffering from that particular symptom. At the end of the diagnosis process, the agent predicts a disease d that the patient may have from the set of all diseases $\mathcal{D}$.

The goal of the agent is to use as few steps as possible while achieving high prediction accuracy. To this end, our prior work in (Tang et al. 2016) employs reinforcement learning (Sutton and Barto 1998) and formulates this problem as a Markov decision process (MDP).

Formally, in time step t, the agent receives a state s_t and then selects an action a_t from a discrete action set $\mathcal{A}$ according to a policy π. In our formulation, $\mathcal{A} = \mathcal{I} \cup \mathcal{D}$. In each time step t, the agent receives a scalar reward r_t. If $a_t \in \mathcal{I}$, the agent performs an inquiry. If the inquiry is repeated, i.e., if $a_t = a_{t'}$ for some $t' < t$, $r_t = -1$ and the interaction is terminated; otherwise $r_t = 0$ and the state is updated according to the response from the patient. If $a_t \in \mathcal{D}$, the agent performs a disease prediction and the interaction is terminated. In this case, $r_t = 1$ if the predicted disease is correct; otherwise $r_t = 0$.

The goal is to find an optimal policy such that the agent maximizes the expected discounted total reward, i.e., the expected return $R_t = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$, where $\gamma \in [0, 1]$ is a discount factor. The state-action Q-value function

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

is the expected return of performing an action a in a state s following a policy π. Given that the Q-value is the sum of an immediate reward and a discounted next-step Q-value, it can be rewritten as a recursive equation:

$$Q^{\pi}(s, a) = \mathbb{E}_{s'}[r + \gamma \mathbb{E}_{a' \sim \pi(s')}[Q^{\pi}(s', a')] \mid s, a, \pi]$$

where s' and a' are the state and action in the next time step, respectively.

The optimal Q-value is the maximum Q-value among all possible policies: $Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$. Also, it can be shown that the optimal Q-value obeys the Bellman equation:

$$Q^*(s, a) = \mathbb{E}_{s'}[r + \gamma \max_{a'} Q^*(s', a') \mid s, a].$$

A policy π is optimal if and only if for every state and action, $Q^{\pi}(s, a) = Q^*(s, a)$. In finite MDPs, such optimal policy always exists. Moreover, an optimal policy can be deduced deterministically by

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a).$$

Thus, some of the reinforcement learning algorithms use a function approximator to estimate the optimal Q-value through training (Mnih et al. 2015; Wang et al. 2016).

Tradeoff between Disease-Prediction Accuracy and Symptom Acquisition

The discount factor γ controls the tradeoff between the number of inquiries and prediction accuracy. Since in our formulation, a correct disease prediction has a reward of 1 and an incorrect disease prediction has a reward of 0, the optimal Q-value of a disease-prediction action ($a \in \mathcal{D}$) is the probability of a patient having the corresponding disease. On the other hand, the optimal Q-value of an inquiry action ($a \in \mathcal{I}$) equals the current step reward plus the discounted expected future rewards (by the Bellman equation). When the Q-value is optimal, the current step reward must be equal to 0 because no repeated action is occurred. Therefore, the net Q-value of an inquiry action is the discounted expected future rewards. This value can also be regarded as the "discounted prediction accuracy."

Thus, when we choose an action based on the Q-values (of $\mathcal{I}$ and $\mathcal{D}$), the discounted prediction accuracies (for inquiry actions) and the prediction accuracies (for disease-prediction actions) are compared. From this perspective, performing more inquiries may result in a higher accuracy in the future but such potential is penalized by the discount factor γ. As a consequence, a disease-prediction action may be chosen instead of an inquiry action.

Anatomical Ensemble

To reduce the problem space, we can divide a human body into parts and the possible symptoms of each part is then much reduced to conduct inferences. There are at least two ways to perform this divide-and-conquer: by medical systems and by anatomical parts. Hospitals divide a body into systems including nervous, circulatory, lymphatic urinary, reproductive, respiratory, digestive, skin/integumentary, endocrine, and musculoskeletal. However, such division is not comprehensible by a typical user. Therefore, our prior work (Tang et al. 2016) devised our model to be an ensemble of different anatomical part models $\mathcal{M} = \{m_p \mid p \in \mathcal{P}\}$. There are eleven anatomical parts $\mathcal{P} = \{$*abdomen*, *arm*, *back*, *buttock*, *chest*, *general symptoms*, *head*, *leg*, *neck*, *pelvis*, *skin*$\}$. The model m_p of each anatomical part $p \in \mathcal{P}$ is responsible for a subset of diseases $\mathcal{D}_p \subseteq \mathcal{D}$. Similarly, we denote the subset of the symptoms associated with m_p by $\mathcal{I}_p \subseteq \mathcal{I}$. Note that the disease sets as well as symptom sets may overlap between different parts. For example, the disease *food allergy* can happen in parts *neck*, *chest*, and *abdomen*.

A neural network is employed as the Q-value estimator for each anatomical part model m_p. The action set of m_p is $\mathcal{A}_p = \mathcal{I}_p \cup \mathcal{D}_p$. The state s^p for m_p is a combination of related symptom statuses. A symptom can be in one of three statuses: *true*, *false*, and *unknown*. The status of a symptom is *true* if the patient has responded positively about the symptom or the symptom is the initial symptom. If the patient has responded negatively about a symptom, then the status of that symptom is *false*. Otherwise, the status is *unknown*. We use a three-element one-hot vector[2] b_i to encode the status of a symptom i. Formally, the state of m_p is $s^p = [b_1^T, b_2^T, \ldots, b_{|\mathcal{I}_p|}^T]^T$, i.e., the concatenation of one-hot encoded statuses of each symptom. Moreover, we denote the policy of each anatomical part model m_p by π_{m_p}.

Our prior work (Tang et al. 2016) chose one anatomical part model based on the initial symptom and the model was used throughout the whole diagnosis process. This approach can bring about several issues. For example, it is possible that the target disease does not belong to the disease set of the chosen anatomical part. In addition, if only a single anatomical part is considered, the other anatomical parts are not fully utilized. In the subsequent sections, we propose remedies to address these issues.

Hierarchical Reinforcement Learning

To address the issue of fixing on one anatomical part, we propose a master agent to assemble models of anatomical parts. The main idea is to imitate a group of doctors with different expertise who jointly diagnose a patient. Since a patient can only accept an inquiry from one doctor at a time, a host is required to appoint doctors in turn to inquire the patient. The master agent in our model acts like the host. At each step, the master agent is responsible for appointing an anatomical part model to perform a symptom inquiry or a disease prediction. This approach essentially creates a hierarchy in our model by introducing a level of abstraction since the master agent cannot directly perform inquiry and prediction actions. Instead, the master agent treats those anatomical part models as subroutines, and the duty of the master agent is to pick one of the anatomical part models at each time step.

The concept of this two-level hierarchy can be described more precisely using reinforcement learning terms introduced in the previous section. The first hierarchy level is a master agent M. The master M possesses its action space $\mathcal{A}_M$ and policy π_M. In this level, the action space $\mathcal{A}_M$ equals $\mathcal{P}$, the set of anatomical parts. At step t, the master agent enters state s_t, and it picks an action a_t^M from $\mathcal{A}_M$ according to its policy π_M. The second level of hierarchy consists of anatomical models m_p. If the master performs action a^M, the task is delegated to the anatomical model $m_p = m_{a^M}$. Once the model m_p is selected, the actual action $a_t \in \mathcal{A}$ is then performed according to the policy of m_p, denoted as π_{m_p}. With this two-level abstraction, our model $\mathcal{M}$ is denoted as $\mathcal{M} = \{M\} \cup \{m_p \mid p \in \mathcal{P}\}$.

In the literature of hierarchical reinforcement learning, there is a framework called *option* (Sutton, Precup, and Singh 1999). An *option* $\langle I, \pi, \beta \rangle$ contains three components: I is a set of states where this option is available, π is the policy of this option, and β determines the probability of terminating the option in a given state. In this framework, a master agent selects an option among all available options in the current state. Then the chosen option gets to execute for a number of steps according to its policy π. In each time step we sample on $\beta(s)$ with the current state s to determine whether this option should be terminated. Once an option has been terminated, the master agent selects a next option to execute. Our approach can be viewed as a simplified version of *option*. Each anatomical part model m_p can be regarded

[2] A vector $v \in \mathbb{B}^n$ is one-hot if $\sum_j v_j = 1$.

Name	Type	Input Size	Output Size
FC1	Linear + ReLU	$\|\mathcal{I}\| \times 3$	$1024 \times \omega$
FC2	Linear + ReLU	$1024 \times \omega$	$1024 \times \omega$
FC3	Linear + ReLU	$1024 \times \omega$	$512 \times \omega$
FC4	Linear	$512 \times \omega$	$\|\mathcal{P}\|$

Table 2: The network architecture of our master model.

as an *option* $\langle I, \pi, \beta \rangle$. The input set I is the set of all possible states because every anatomical part model is available in all states. The policy π corresponds to π_{m_p}. The termination condition β always evaluates to be 1 since our master model re-selects an anatomical model for each step.

Model

As stated in the previous section, an optimal policy can be obtained through an optimal Q-function. Therefore, to find the optimal policy, one approach is to find an optimal Q-function. One challenge of this approach is that the state and action space are usually high in their dimensions. To address this issue, Mnih et al. proposed a deep Q-network (DQN) architecture as a function approximator for Q-functions.

We adopt DQN as a model to approximate the Q-function of the master agent M. Given a state s, the output layer of the master model outputs a Q-value for each action $a \in \mathcal{A}_M$. At each step, the master model can pick an anatomical model m_p according to the Q-values of the master model. The model $m_p = m_{a^M}$ is selected when its corresponding action a^M has the maximum Q-value among all actions.

The master model consists of four fully connected (FC) layers. The rectified linear units (ReLUs) (Nair and Hinton 2010) are used after each layer except for the last. The width of each layer is shown in Table 2. Note that since the size of $\mathcal{I}$ varies across our experimental tasks, to cope with these changes, we can adjust the width of each hidden layer by a linear factor ω.

Training

Individual anatomical part models are first trained by the method of (Tang et al. 2016). Then, the master model can be trained after individual anatomical part models have been trained because training the master requires the inference results of the parts.

To train the master model, we use the DQN training algorithm (Mnih et al. 2013). The loss function computes the squared error between the Q-value output from the network and the Q-value obtained through the Bellman equation, which can be written as

$$L_j(\theta_j) = \mathbb{E}_{s,a,r,s'}[(y_j - Q(s, a; \theta_j))^2], \qquad (1)$$

where target $y_j = r + \gamma \max_{a'} Q(s', a'; \theta^-)$ is evaluated by a separate *target network* $Q(s', a'; \theta^-)$ with parameters θ^- (Mnih et al. 2015). The variable j is the index of training iteration. Note that if action a terminates the interaction, $y_j = r$. To evaluate the expectation in the loss function, we sample a batch of (s, a, r, s') tuples and use the mean square error as an approximation. To improve training stability and convergence, the target network is fixed for a number of training iterations before θ^- is updated to be θ. The parameters θ can be updated by the standard backward propagation algorithm.

Algorithm 1 details the training algorithm of our master model. Before the interactive process starts, the state s is

Algorithm 1: TrainingMasterModel

Input : $\{m_p \mid p \in \mathcal{P}\}$ // Set of anatomical models
$\{\mathcal{I}_p \mid p \in \mathcal{P}\}$ // Set of symptom sets
$\mathcal{A}_M$ // Action set of the master model
$\mathcal{D}_M$ // Disease set of the master model
ϵ // Epsilon greedy parameter
γ // Discount factor
δ // Termination threshold
Output : θ // Parameters of the master model
Variable: $x, target$; // Data and ground truth
$s, a, r, s', a^M, cp, s^{cp}$;
$\theta^-, y, loss$;
$\mathcal{H}$; // Inquiry history

begin
  $x, target \longleftarrow DataSampler()$
  $s \longleftarrow InitializeState(x)$
  $\mathcal{H} \longleftarrow \phi$
  $loss \longleftarrow \infty$
  while $loss > \delta$ **do**
    if $UniformSampler([0, 1)) < \epsilon$ **then**
      $a^M \longleftarrow UniformSampler(\mathcal{A}_M)$
    else
      $a^M \longleftarrow \arg\max_a Q_M(s, a; \theta)$
    end
    $cp \longleftarrow a^M$
    $s^{cp} \longleftarrow ExtractState(s, \mathcal{I}_{cp})$
    $a \longleftarrow \arg\max_a Q_{m_{cp}}(s^{cp}, a)$
    $r \longleftarrow \begin{cases} -1, & \text{if } a \in \mathcal{H} \\ 1, & \text{if } a = target \\ 0, & \text{otherwise} \end{cases}$
    if $a \in \mathcal{D}_M$ *or* $a \in \mathcal{H}$ **then**
      $x, target \longleftarrow DataSampler()$
      $s' \longleftarrow InitializeState(x)$
      $\mathcal{H} \longleftarrow \phi$
      $y \longleftarrow r$
    else
      $s' \longleftarrow UpdateState(s, x, a)$
      $\mathcal{H} \longleftarrow \mathcal{H} \cup \{a\}$
      $y \longleftarrow r + \gamma \max_{a'} Q_M(s', a'; \theta^-)$
    end
    $loss \longleftarrow (y - Q_M(s, cp; \theta))^2$
    $\theta \longleftarrow GradientUpdate(\theta, loss)$
    $\theta^- \longleftarrow \theta$ for every C iterations
    $s \longleftarrow s'$
  end
  return θ
end

initialized based on the initial symptom of a training example. In the initialized state s, except for the initial symptom being *true*, all other symptoms are *unknown*. To begin a training iteration, we first infer the master action a^M, which is essentially the anatomical part selected to be used in this iteration. In training time, the balance of exploration (exploring unseen states) and exploitation (utilizing learned knowledge to select the best action) is important. We choose to use epsilon greedy to cope with this: With probability ϵ, the master action a^M is picked uniformly from $\mathcal{A}_M$; otherwise, a^M is assigned to the best action learned so far, i.e., $a^M = \arg\max_{a \in \mathcal{A}_M} Q_M(s, a)$.

The next step is to infer the action of previously selected anatomical model. For annotation simplicity, we shall denote the chosen part as $cp = a^M$, and thus the selected anatomical model is m_{cp}. The state s^{cp} used by m_{cp} is different from the state s used by M. This is because the symptom set $\mathcal{I}_{cp}$ is a subset of $\mathcal{I}_M$, the one used by the master model. We can obtain s^{cp} by extracting relevant symptoms from s. Therefore, the inferred action given s^{cp} is $a = \arg\max_{a \in \mathcal{A}_{cp}} Q_{m_{cp}}(s^{cp}, a)$.

With the action a emitted by m_{cp}, we can interact with the patient and update the state and the master model. In order to update the state, the response of the patient is obtained and the next state s' is updated accordingly. However, if a is repeated or $a \in \mathcal{D}$, the interaction with the present patient is terminated. In this case, the next state s' is set to a new initial state created by a newly sampled patient. After the state is updated, we can use Equation 1 to update the master model. Note that a in Equation 1 is actually a^M when we update the master model.

The algorithm described above is the training procedure for one example. However, using stochastic gradient descent with one example can result in unstable gradients. We use a batch of parallel patients to overcome this problem. At each step of training, each patient independently receives an inquiry or diagnosis and maintains its own state. Since the length of the interaction is not fixed, each patient can finish its interaction at a different time. When the interaction of a certain patient is terminated, we can replace the old patient with a newly sampled one on-the-fly. Therefore, the number of inquiries taken by each patient within a batch can be different. This makes a training batch more diverse and uncorrelated, resulting in a similar effect of replay memory (Mnih et al. 2013).

Modeling Context

To model context, we can modify the underlying MDP and state representation. The state is augmented with an extra encoding of contextual information, i.e., $s = [b^T, c^T]^T$, where b denotes the symptom statuses capturing the inquire history of the interaction and c denotes the contextual information possessed by a patient. Given a state $s = [b^T, c^T]^T$, our master model M outputs a Q-value of each action $a \in \mathcal{A}_M$.

More specifically, the encoding scheme of b is the same as the original one. The newly enhanced part is the contextual information c that currently comprises the *age*, *gender*, and *season* information of a patient. (Any other *who*, *when*, and *where* information can be easily incorporated.) Here, we denote $c = [c_{age}^T, c_{gender}^T, c_{season}^T]^T$. First, the age information $c_{age} \in \mathbb{N}$ is useful because some diseases have higher possibilities on babies whereas some have higher possibilities on adults. For example, meningitis typically occurs on children, and Alzheimer's disease on the elderly. Second, the gender information $c_{gender} \in \mathbb{B}$ is important because some diseases strongly correlate with gender. For example, females may have problems in uterus, and males may have prostate cancer. Third, the season information $c_{season} \in \mathbb{B}^4$ (a four-element one-hot vector) is also helpful because some diseases (e.g., those transmitted by mosquitoes such as malaria, dengue, filariasis, chikungunya, yellow fever, and Zika fever) are associated with seasons.

Given the new state representation, our algorithm requires to be modified slightly. In our definition, each action a has two types: an inquiry action ($a \in \mathcal{I}$) or a diagnosis action ($a \in \mathcal{D}$). If the maximum Q-value of the outputs corresponds to an inquiry action, then our model inquires the corresponding symptom to a user, obtains a feedback, and proceeds to the next time step. The feedback is incorporated into the next state $s_{t+1} = [b_{t+1}^T, c^T]^T$ according to our symptom status encoding scheme. Otherwise, the maximum Q-value corresponds to a diagnosis action. In the latter case, our model predicts the maximum-Q-value disease and then terminates.

Context-Aware Policy Transformation

Previously, the model directly takes the contextual information into consideration. In this subsection, we propose an alternative direction. Given an optimal policy π^*, which does not consider context, can we transform it to an optimal policy π_c^*, which does consider context? We call this approach the *context-aware policy transformation*. We shall prove this transformation holds under certain assumptions.

Proposition 1. *Let d, s, and c denote disease, symptom, and context, respectively. If we assume s and c are conditionally independent given d, then*

$$p(d \mid s, c) = \frac{p(d \mid s)p(c \mid d)}{p(c \mid s)}.$$

Proof.

$$\begin{aligned} p(d \mid s, c) &= \frac{p(s, c \mid d)p(d)}{p(c \mid s)p(s)} \\ &= \frac{p(s \mid d)p(c \mid d)p(d)}{p(c \mid s)p(s)} \\ &= \frac{p(d \mid s)p(c \mid d)}{p(c \mid s)} \end{aligned}$$

□

Now, let Q_c^* denote the optimal value function considering context and Q^* the optimal value function without considering context. We have the following lemma.

Lemma 2. *If $\pi_c^*(s) \in \mathcal{D}$, then*

$$\pi_c^*(s) = \arg\max_{a \in \mathcal{D}} Q^*(s, a)p(c \mid a).$$

Proof. If $\arg\max_a Q_c^*(s,a) \in \mathcal{D}$, then

$$\begin{aligned}\arg\max_a Q_c^*(s,a) &= \arg\max_{a\in\mathcal{D}} \mathbb{E}[\mathbb{1}_{a=y} \mid s,c] \\ &= \arg\max_d p(d \mid s,c) \\ &= \arg\max_d p(d \mid s)p(c \mid d) \\ &= \arg\max_{a\in\mathcal{D}} Q^*(s,a)p(c \mid a).\end{aligned}$$

□

From Lemma 2, we can see that if the action a chosen from π_c^* is a diagnosis action ($a \in \mathcal{D}$), 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 $p(c \mid d)$. Next, we analyze another case when the action is an inquiry action.

Lemma 3. *Assume $\gamma = 1$. If $\pi_c^*(s) \in \mathcal{I}$, then*

$$\pi_c^*(s) \approx \arg\max_{a\in\mathcal{I}} Q^*(s,a)p(c \mid s')\frac{p(\hat{s'} \mid s,c,a)}{p(\hat{s'} \mid s,a)}.$$

Proof. Let y be the target disease. If $\arg\max_a Q_c^*(s,a) \in \mathcal{I}$, then

$$\begin{aligned}\arg\max_a Q_c^*(s,a) &= \arg\max_{a\in\mathcal{I}} \mathbb{E}_{s'\sim p(s'|s,c,a)}[p(y \mid s',c)] \\ &= \arg\max_{a\in\mathcal{I}} \mathbb{E}_{s'\sim p(s'|s,c,a)}\left[\frac{p(y \mid s')p(c \mid y)}{p(c \mid s')}\right] \\ &= \arg\max_{a\in\mathcal{I}} \mathbb{E}_{s'\sim p(s'|s,a)}\left[\frac{p(s' \mid s,c,a)}{p(s' \mid s,a)}\frac{p(y \mid s')}{p(c \mid s')}\right] \\ &\approx \arg\max_{a\in\mathcal{I}} \frac{p(\hat{s'} \mid s,c,a)}{p(\hat{s'} \mid s,a)p(c \mid s')}Q^*(s,a).\end{aligned}$$

□

Lemma 3 states that when π_c^* selects actions from $\mathcal{I}$, the transformation will require three probability distributions $p(c \mid s)$, $p(s' \mid s,a)$, and $p(s' \mid s,c,a)$. From Proposition 1, we have

$$\begin{aligned}p(c \mid s) &= \sum_{d\in\mathcal{D}} p(d \mid s)p(c \mid d) \\ &= \sum_{d\in\mathcal{D}} Q^*(s,d)p(c \mid d).\end{aligned}$$

In practice, $p(c \mid d)$ can be available and therefore $p(c \mid s)$ can also be available. However, the other two distributions $p(s' \mid s,a)$ and $p(s' \mid s,c,a)$ are the transitions of MDPs with and without context which may not be available.

Remark 1. *Although the theoretical result of policy transformation from π^* to π_c^* is established, in practice, the MDP transitions $p(s' \mid s,a)$ and $p(s' \mid s,c,a)$ may not be available, and γ may be unequal to 1. In these cases, we can still use Lemma 2 to transform the disease-prediction probability in the last diagnosis step.*

Task	$\lvert\mathcal{D}_p\rvert$	$\lvert\bigcup_p \mathcal{D}_p\rvert$	$\lvert\bigcup_p \mathcal{I}_p\rvert$	ω
Task 1	25	73	246	1
Task 2	50	136	302	2
Task 3	75	196	327	3
Task 4	100	255	340	4

Table 3: The settings of our four experimental tasks.

Experiments

Medical data is difficult to obtain and share between researchers because of privacy laws (e.g., the Health Insurance Portability and Accountability Act; HIPAA) and security concerns. While there are some publicly available electronic health record (EHR) datasets, these datasets often lack symptom-related information. For example, the MIMIC-III dataset (Johnson et al. 2016) was collected at intensive care units without full symptom information. To evaluate our algorithm, we generated simulated data based on SymCAT's symptom-disease database (AHEAD Research Inc 2017) composed of 801 diseases.

Each disease in SymCAT is associated with its symptoms and probabilities, i.e., $p(s \mid d)$. We further cleaned up the set of diseases by removing the ones that do not appear in the Centers for Disease Control and Prevention (CDC) database (Centers for Disease Control and Prevention 2017) and the ones that are logical supersets of some of the other diseases indicated in the UMLS medical database (National Institutes of Health 2017). The resulting probability database consists of 650 diseases.

Next we assembled four sub-datasets for four experimental tasks, each containing a different number of diseases. With the aid of experts, we manually classified each disease into one or more anatomical parts. For each anatomical part, we reserved its top 25, 50, 75 and 100 diseases in terms of the number of occurrences in the CDC records. Table 3 shows the detailed numbers of our four tasks. The five columns depict the task name, the number of diseases in each anatomical part, the number of diseases in the union of all anatomical parts, the number of symptoms in the union of all anatomical parts, and the value of parameter ω in our network. Note that a disease may occur in more than one anatomical part, which is the reason why the number of diseases in the union set is less than the sum of diseases in all parts.

For training, we generated simulated patients dynamically by the following process. We sampled a target disease uniformly. Then for each associated symptom, we sampled a Boolean value indicating whether the patient suffers form that symptom using a Bernoulli distribution with the probability taken from the database. For those symptoms that are not associated with the chosen disease, we performed the same sampling process with a probability of 0.0001 to introduce a floor probability so as to mitigate the negative effect of a erroneous response. If the given patient does not have any symptoms, the symptom generation process starts from scratch again. Otherwise, an initial symptom is picked uniformly among all sampled symptoms. In all tasks, we used

Tasks	Task 1		Task 2		Task 3		Task 4	
	Best Prior Work (Tang et al. 2016)	Hierarchical Model	Best Prior Work (Tang et al. 2016)	Hierarchical Model	Best Prior Work (Tang et al. 2016)	Hierarchical Model	Best Prior Work (Tang et al. 2016)	Hierarchical Model
Top 1	48.12 ± 0.15	63.55 ± 0.15	34.59 ± 0.11	44.50 ± 0.11	25.46 ± 0.08	32.87 ± 0.09	21.24 ± 0.07	26.26 ± 0.07
Top 3	59.01 ± 0.15	73.35 ± 0.13	41.58 ± 0.11	51.90 ± 0.11	29.63 ± 0.08	38.02 ± 0.09	24.56 ± 0.07	29.81 ± 0.07
Top 5	63.23 ± 0.15	77.94 ± 0.13	45.08 ± 0.11	55.03 ± 0.11	31.82 ± 0.09	40.20 ± 0.09	26.15 ± 0.07	31.42 ± 0.07
#Steps	7.17 ± 0.02	7.15 ± 0.01	7.06 ± 0.01	5.73 ± 0.01	5.98 ± 0.01	5.14 ± 0.00	6.94 ± 0.01	5.01 ± 0.00

Table 4: Experimental results on anatomical model (Tang et al. 2016) and our proposed hierarchical model. The top-n accuracies are shown in percentage with a 99% confidence interval.

Inquiry Stage			
Step	Selected Part	Inquired Symptom	Response
1	General symptoms	Symptoms of prostate	False
2	Chest	Painful urination	False
3	Chest	Side pain	True
4	Back	Fever	False
5	Back	Blood in urine	True
6	Back	Nausea	True
Diagnosis Stage			
Top 5	Disease		
1	**Kidney stone**		
2	Urinary tract infection		
3	Problem during pregnancy		
4	Sprain or strain		
5	Acne		

Table 5: An interaction sequence. The target disease is kidney stone and the initial symptom is frequent urination.

ten million mini-batches, each consisting of 128 samples for training.

We separately produced a testing dataset using the above procedure without generating the floor probability (i.e., the probability for symptoms that are not associated with the chosen disease is 0). We sampled $10,000$ simulated patients for each disease in the testing dataset for each task.

Since training our 11 part models is time-demanding, we used DeepQ Open AI Platform (Zou et al. 2017) to manage our training process. The auto-scaling and task management features of this platform enabled us to conduct our experiments in parallel. Also, the visualization feature helped us monitor the progress of training conveniently.

Table 4 compares the experimental results of our proposed model with the best results of our prior work (Tang et al. 2016), which enjoys the top result published thus far. For each task, four numbers are reported. They are top-1 accuracy, top-3 accuracy, top-5 accuracy, and the average number of inquiries over all symptom-checking interactions. Each of the top-n accuracy numbers represents the percentage of top-n predictions containing the target disease. All numbers are reported along with 99% confidence intervals.

As shown in Table 4, the accuracy of our proposed ensemble scheme is significantly higher than that of the previous model. The average number of symptom inquiries made is also slightly lower. When the number of candidate diseases is small (25 and 50), our model outperforms our prior work by at least ten percentage points in top-1, top-3, and top-5 results. When the number of candidate diseases is large (75 and 100), our accuracy outperforms by at least five percentage points.

#	Without Context	Context-Aware
1	Urinary tract infection	**Kidney stone**
2	**Kidney stone**	Benign blood in urine
3	Benign blood in urine	Venous insufficiency
4	Gastroesophageal reflux disease	Abdominal hernia
5	Venous insufficiency	Metastatic cancer

Table 6: Top-5 diagnosis with/without context. The patient is a man whose age is between 45 and 59 and suffers from kidney stone.

Example

To demonstrate that our master model utilizes the power of different anatomical part models, we display the details of one diagnosis session in Table 5. Note that our prior work (Tang et al. 2016) fails to diagnose this patient. Table 5 shows that our master model first chooses to use *general symptoms*. After three rounds, our master model is able to focus on the most relevant part, *back*, and select that model to produce a correct prediction.

Context-Aware Policy Transformation

We evaluate the context-aware policy transformation by using Lemma 2. The context considered in our experiment includes age and gender[3]. To transform a policy, we are required to evaluate $p(c \mid d)$. Assuming age and gender are independent given that the target disease is known, $p(c \mid d)$ equals $p(age \mid d)p(gender \mid d)$. Note that age is a non-negative real number, which makes the probability continuous and hard to evaluate. To make $p(age \mid d)$ discrete, we quantized ages into several bins, each representing a non-overlapping range of ages. We then obtained those probabilities from SymCAT's disease database. After that, for each simulated patient in the test dataset, we sampled a gender and an age range according to the probabilities $p(gender \mid d)$ and $p(age \mid d)$. We next show some examples demonstrating how context can influence diagnosis, and then pro-

[3]Note that other contextual information such as season can be applied using the same methodology as well.

#	Without Context	Context-Aware
1	Metastatic cancer	**Osteoporosis**
2	Chronic constipation	Metastatic cancer
3	Abdominal hernia	Chronic kidney disease
4	Chronic kidney disease	Decubitus ulcer
5	Gastroesophageal reflux disease	Venous insufficiency

Table 7: Top-5 diagnosis with/without context. The patient is a 75+ woman suffering from osteoporosis.

#	Without Context	Context-Aware
1	**Osteoporosis**	Decubitus ulcer
2	Spondylosis	Venous insufficiency
3	Lumbago	Chronic ulcer
4	Decubitus ulcer	Colorectal cancer
5	Venous insufficiency	Spondylosis

Table 8: Top-5 diagnosis with/without context. The patient is a 75+ man suffering from osteoporosis.

vide the result of an ablation study on the effect of context-aware transformation.

Table 6 demonstrates a case where context refines the result. Without contextual information, the top-1 prediction based solely on the interaction process is inaccurate. If we consider the gender of this patient, we can rule out *urinary tract infection* since it is relatively rare in males. As a result, the target disease *kidney stone* becomes the top-1 after the context-aware transformation.

In Table 7, we show another case in which context fixes the incorrect diagnosis result. If the predictions do not consider context, the top-5 predictions do not include the target disease. When context is considered, *osteoporosis* is boosted since instances of osteoporosis have a distribution that tends towards women rather than men. In this case, the context-aware prediction results in a correct top-1 prediction.

Conversely, in Table 8 we provide a failure case due to context-aware transformation. In this case, osteoporosis is successfully diagnosed without context. However, due to the fact that this patient is a male, which is relatively less often to have osteoporosis, the context-aware transformation misleads the diagnosis by suppressing the probability of osteoporosis.

We further conducted an ablation study to investigate the usefulness of the context-aware policy transformation. In this study, we chose a set of diseases that are influenced by contextual information, and did not include diseases that are evenly distributed among genders and age ranges. The chosen diseases in this study were *problem during pregnancy*, *prostate cancer venous insufficiency*, *actinic keratosis*, *lung cancer*, *skin cancer*, *chlamydia*, and *heart failure*. As examples of contextual influence, problem during pregnancy is only associated with females, while prostate cancer is only attributable to males. Venous insufficiency is unlikely to occur in children. We created another test set that only contains the chosen context-influenced diseases and evaluated the top-5 accuracy of our hierarchical models trained previously. The comparison of hierarchical model and context-aware model on the chosen diseases is presented in Table 9. We can see that the accuracy is improved by at least three percentile with context-aware policy transformation in this ablation study.

Models	Task 1	Task 2	Task 3	Task 4
Hierarchical Model	76.16	57.51	32.18	33.76
Context-Aware Model	83.62	63.37	36.58	37.96

Table 9: An ablation study of context-aware policy transformation and a comparison of top-5 accuracy on hierarchical and context-aware model.

Conclusion

We have shown that the proposed master model can orchestrate anatomical part models to jointly perform symptom checking. This hierarchical ensemble scheme significantly improves diagnosis accuracy compared to our prior work (Tang et al. 2016), while making a similar or fewer number of symptom inquiries.

When considering contextual information, we have shown that there are two different ways to integrate contextual information. One way is to treat context as an input to the model, the other is using context-aware transformation. We demonstrated the benefit of such transformation by some qualitative cases when the result can be refined by strong hints from context.

Figure 1: XPRIZE DeepQ Tricorder. DeepQ consists of four compartments. On the top is a mobile phone, which runs a symptom checker. The drawer on the right-hand-side contains optical sense. The drawer on the lower-front contains vital sense and breath sense. The drawer on the left-hand-side contains blood/urine sense. The symptom checker guides a patient on which tests to conduct.

In our future work, we plan to further improve accuracy by exploring three approaches.

- Incorporating more information for diagnosis. As we stated in the introduction section, without information such as medical records, family history, and lab tests, symptoms alone cannot achieve the optimal diagnosis accuracy. Therefore, our first logical step is to incorporate more information into our model. We will also actively seek for or develop real-world datasets that we can use to conduct practical experiments.
- Suggesting lab tests before diagnosis. We can use the symptom checker as a tool to suggest collecting missing information when it can improve diagnosis accuracy. For instance, when our model cannot decide between two diseases, it can suggest lab tests to provide missing information for disambiguation. Once additional useful information has been collected, we believe that diagnosis accuracy will be further improved. In our XPRIZE DeepQ Tricorder device (Chang et al. 2017), we indeed employ a symptom checker (see Figure 1) to suggest lab tests for a patient to conduct before a diagnosis.
- Experimenting with other latent layers. In this paper, we define our latent layer by separating diseases into different body parts. The user interface of our DeepQ Tricorder (Chang et al. 2017) benefits from this body-part-grouping method. There are also other potential ways to define our latent layer, such as grouping by systems (e.g., digestive, nervous, circulatory, lymphatic urinary, reproductive, respiratory, and digestive). We believe that pursuing the best latent layer using a data-driven approach could be a promising direction of research.

Acknowledgements

We would like to thank the following individuals for their contributions. We thank Jocelyn Chang and Chun-Nan Chou for their help in proofreading the paper. We thank Chun-Yen Chen, Ting-Wei Lin, Cheng-Lung Sung, Chia-Chin Tsao, Kuan-Chieh Tung, Jui-Lin Wu, and Shang-Xuan Zou for their help in running the experiments of this paper. We also thank Ting-Jung Chang and Emily Chang for providing their medical knowledge.

References

AHEAD Research Inc. 2017. SymCAT: Symptom-based, computer assisted triage. `http://www.symcat.com`.

Centers for Disease Control and Prevention. 2017. Ambulatory health care data. `https://www.cdc.gov/nchs/ahcd/index.htm`.

Chang, E. Y.; Wu, M.-H.; Tang, K.-F.; Kao, H.-C.; and Chou, C.-N. 2017. Artificial intelligence in XPRIZE DeepQ tricorder. In *MM Workshop on Multimedia and Health*.

Hayashi, Y. 1991. A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis. In *Advances in Neural Information Processing Systems 3*. 578–584.

Johnson, A. E.; Pollard, T. J.; Shen, L.; wei H Lehman, L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. In *Scientific data*, volume 3.

Kohavi, R. 1996. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, 202–207.

Kononenko, I. 1993. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence* 7(4):317–337.

Kononenko, I. 2001. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine* 23(1):89–109.

Ledley, R., and Lusted, L. 1959. Reasoning foundations of medical diagnosis symbolic logic, probability, and value theory aid our understanding of how physicians reason. *Science* 130(3366):9–21.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing Atari with deep reinforcement learning. *CoRR* abs/1312.5602.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of The 27th International Conference on Machine Learning*, 807–814.

National Institutes of Health. 2017. Unified medical language system. `https://www.nlm.nih.gov/research/umls/`.

Qualcomm. 2017. Xprize Tricorder Winning Teams. `http://tricorder.xprize.org/teams`.

Semigran, H. L.; Linder, J. A.; Gidengil, C.; and Mehrotra, A. 2015. Evaluation of symptom checkers for self diagnosis and triage: audit study. *BMJ* 351.

Sutton, R., and Barto, A. 1998. *Reinforcement learning: An introduction*, volume 116. Cambridge Univ Press.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

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

Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; and Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, 1995–2003.

Zou, S.-X.; Chen, C.-Y.; Wu, J.-L.; Chou, C.-N.; Tsao, C.-C.; Tung, K.-C.; Lin, T.-W.; Sung, C.-L.; and Chang, E. Y. 2017. Distributed training large-scale deep architectures. In *Proceedings of The 13th International Conference on Advanced Data Mining and Applications*, 18–32.