Building Conversational Diagnosis Systems for Fine-grained Diseases using Few Annotated Data

Yiping Song[1], Wei Ju[2], Zhiliang Tian[1], Luchen Liu[2], Ming Zhang[2]*, and Zheng Xie[1]

[1] National University of Defense Technology
{songyiping,tianzhiliang,xiezheng81}@nudt.edu.cn
[2] Peking University {juwei,liuluchen,mzhang_cs}@pku.edu.cn

Abstract. The conversational diagnosis system aims to interview patients and make the diagnosis just like doctors do. Most existing methods rely on medical dialog corpora collected from various medical forums. Compared to experts' annotated data, the dialog corpora are easily accessible, but lack medical knowledge and diagnostic decision logic. Thus, those systems can only handle coarse-grained diseases but achieve poor performance on fine-grained diseases. In this paper, we present a Reinforcement Learning (RL) framework that leverages a few annotated (from experts) and unannotated (from online forums) dialogs to make the diagnosis for fine-grained diseases. We summarize the doctor's diagnosis logic from the unannotated dialogs, then build a user simulator by annotated dialogs for RL training. In this way, a few annotated data are sufficient to support fine-grained disease diagnosis with the assistance of unannotated data. The experiments on eight fine-grained diseases show that our approach outperforms other competitive baselines.

Keywords: Diagnosis Systems · Fine-grained Diseases · Few Data

1 Introduction

The conversational diagnosis system [23, 26, 16], which is a typical task-oriented dialog system [27, 22] in medical domain. It enables a smooth interaction between patients and computers via natural language and provides a reliable diagnosis for patients without too much time and expense. Existing conversational diagnosis systems usually employ reinforcement learning (RL) methods to build models for coarse-grained diseases using the dialog corpora from online forums. The training data from online forums are cheap and easy to access. Such dialogs contain medical common sense so the existing systems can do well on some easily diagnosed diseases, such as upper respiratory infection and infantile diarrhea. These diseases are from different clinical departments and can be easily diagnosed by some obvious symptoms, and we call them coarse-grained diseases. For example, DX dataset [23, 26], the only available conversational diagnosis dataset, consists of four types of diseases from different clinical departments.

* corresponding author

Compared with coarse-grained diseases, fine-grained diseases, where the diseases belong to the same clinical department, are of more importance in the real-world scenario. The reason is that people are more likely to seek help for a problem that cannot be solved with intuition. For fine-grained diseases, more training data is required because the differences between such diseases are so minor, and these diseases usually relate to fine-grained symptoms with various attributes. Moreover, diagnosing fine-grained diseases does not simply rely on the co-occurrence of several symptoms as the coarse-grained disease, but requires more medical knowledge and reliable decision logic from professional doctors. However, online dialogs, which may be written by laymen, cannot provide such professional and reliable information, so annotation from experts is vital for fine-grained diagnosis. Expert annotation in the medical domain is so expensive and time-consuming, and as a result, the insufficiency of annotated data becomes a bottleneck for building fine-grained disease diagnosis systems.

To remedy the lack of annotated data, we can exploit unannotated data. Particularly, we can extract and leverage the general diagnosis logic from the online dialogs, since the diagnosis logics on different diseases are similar and do not require exact medical knowledge. For example, when netizen A claims he has a fever, another netizen B may ask whether he or she has taken vaccines. No matter whether the final diagnosis is correct or not, this partial inquiry logic can be utilized to guide the policy of our diagnosis system. However, these online dialogs need to be used in a proper way because netizens are unprofessional.

In this paper, we propose to leverage a few expert annotated data and unannotated data from an online medical forum to train the fine-grained conversational diagnosis system. Our system is under the RL framework. For unannotated data, we extract the diagnostic logic from dialog sessions, and store them into the experience pool in the form of "state-action-next state-reward" quadruples. Even though the logic may not be accurate, it can be verified by the interaction under the RL framework. For annotated data, we build a user simulator to provide an environment for RL training. The simulator produces feedback to the system by imitating the annotated data, while the agent interacts with the simulator to learn an optimal policy under the guidance of the experience pool.

Our contributions are three-fold: (1) We are the first one to build an efficient conversational diagnosis system for fine-grained diseases which belong to the same clinical department. (2) We propose an RL-based framework that leverages a few annotated data and large-scaled unannotated data. (3) We extract diagnostic logic catering to the real doctors' diagnosis inference from unannotated data, and the logic is interpretable and proves to be useful in diagnosis.

2 Related Work

Task-oriented Conversation Systems. The Task-oriented conversation system [20, 11] aims to accomplish a given task in a vertical domain. Existing works cover a limited number of domains including guiding, shopping, and education [5, 7, 21, 8, 27]. The approaches to building the task-oriented systems have two categories: the modular-based [26, 23, 24] and end-to-end trainable methods [25, 3]. In this paper, we use the former one. Insufficiency of the corpus is a typical bottleneck for task-oriented conversation systems. Some works use domain adaptation [8, 2], multi-task [15, 6], few-shot learning [27, 19, 14] and transfer learning [1, 4] technique. But they require multiple similar domains, but we cannot find similar ones for the medical domain. Another solution [17] is to use a self-play strategy to create new annotated data, but it requires crowdsourcing to correct the created dialogs, which is money-consuming. Li et al. [13] propose a probabilistic framework that can take advantage of expert knowledge, and can help to analyze fault cases. But this is not a purely data-driven method and highly relies on domain experts.

Diagnosis Systems. Automatic diagnosis systems are of great significance [9, 18]. Tang et al. [20] use a symptom-checking list that allows the doctor to inquire about the symptoms and make the diagnosis, which can be regarded as a prototype of conversation systems. Kao et al. [11] build a system that takes the personal information of patients into consideration, but personal information is hard to collect due to the privacy policy. Wei et al. [23] are the first to formulate the inquiring process as a form of dialogue. Natural language is used for interaction, and many other actions such as greetings are allowed in the conversation to achieve a better user experience. They build the model under the RL framework, where the system works as the agent and a patient simulator works as the environment. Deep Q-network (DQN) is used to train the policy of the system. This is also the base model of our work. Xu et al. [26] propose to use two branches to predict a probability vector over all possible actions separately. One branch uses a DQN model like [23], and the other uses the statistical co-occurrence frequency of the symptoms and diseases. Then the two probability vectors are added together to make the final decision. Luo et al. [14] propose to learn the disease embedding by encoding the conversation histories in the training set and use the embedding to predict action at each conversational turn. The above models use the corpus from online forums to build the patient simulator and are only applicable to the diseases that can be diagnosed by common sense. While for fine-grained diseases, more medical knowledge and decision logic are required, so the corpus must be annotated by doctors instead of the netizens. Since the annotation from doctors is expensive, our work proposes an RL-based method for fine-grained diseases which only need a few annotated data.

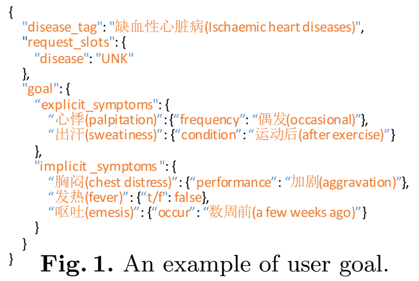
3 Methodology

We first introduce the RL architecture, then explain how to use two different sources of data to train the dialog management module in two steps: diagnostic logic extraction and ensemble training. We describe the data flows in the end.

3.1 Reinforcement Learning (RL) Framework

Under the RL-based framework, the proposed conversational diagnosis system acts as the doctor, which is the agent in RL settings. A patient simulator acts as the patient, which is the environment in RL settings. The patient simulator first elaborates the illness condition, then the system continually interacts with the patient simulator to collect useful information (symptoms) to get a comprehensive understanding of his or her illness condition, then makes the final diagnosis at the end of the conversation. Following the settings of RL, other essential components are as follows.

State s. s_t records all behaviors of the system and patient until the current turn t. It contains all the symptoms with the attributes inquired by the system. **Action** a. a_t is agent's action at t-th turn. Each action has two types: the request action that inquires about the symptom, the inform action that makes a diagnosis. **Reward** r. The reward r is the immediate feedback given by the simulator after the system taking action a. The reward is $+44$ if the system makes the right diagnosis within the maximal turn, and the reward is $+2$ if the system hits a symptom or attribute in the user goal. Otherwise, the reward is -1 to encourage shorter conversation. **Policy** π. Policy π defines how the system acts in accordance with the current state s, noted as $\pi(a_t|s_t)$.



```
{
  "disease_tag": "缺血性心脏病(Ischaemic heart diseases)",
  "request_slots": {
    "disease": "UNK"
  },
  "goal": {
    “explicit_symptoms": {
      “心悸(palpitation)”: {“frequency”: “偶发(occasional)”},
      “出汗(sweatiness)”: {“condition”: “运动后(after exercise)”}
    },
    "implicit _symptoms ": {
      “胸闷(chest distress)”: {“performance”: “加剧(aggravation)”},
      “发热(fever)”: {“t/f": false},
      “呕吐(emesis)”: {“occur”: “数周前(a few weeks ago)”}
    }
  }
}
```

Fig. 1. An example of user goal.

Patient simulator. The behaviors of the patient simulator during the interaction are based on the "user goal" (Fig. 1) extracted from the Electronic Health Record (EHR). It consists of 3 parts: the patient's disease, explicit symptoms, and implicit symptoms. The disease tag is the ground truth of the patient's disease, which is expected to be diagnosed by the system. The explicit symptoms are informed to the system at the beginning of the conversation. The implicit symptoms are other useful symptoms that the system needs to inquire about for diagnosis. Different from [23], we define several attributes for each symptom to better describe the patient's condition. The attributes are a binary variable (indicating whether the patient has this symptom), frequency (how frequent the symptom happens), duration (how long the symptom lasts), severity (how serious the symptom is), and condition (under which condition the symptom happens). These attributes are essential for fine-grained disease diagnosis. It is because the co-occurrence of symptoms and diseases is no longer sufficient to diagnose fine-grained diseases as the coarse-grained ones, while the differences lay in attributes. Notice that various attributes enlarge the action space for the system, making policy learning more challenging.

Diagnosis system. We use a modular-based framework to build the system, which consists of a Natural Language Understanding (NLU) module, a Dialog Management (DM) module and a Natural Language Generation (NLG)

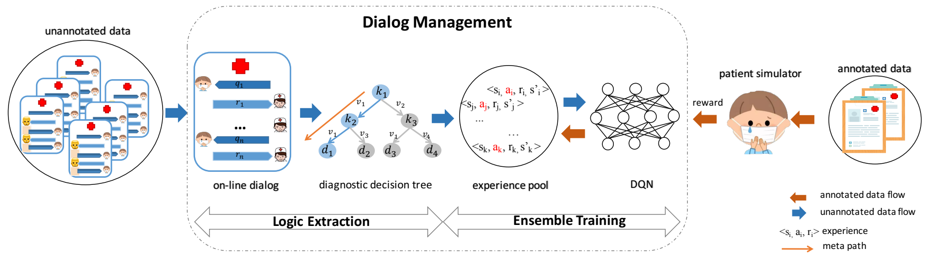


Fig. 2. The Dialog Management module of the proposed model where both annotated data (in orange arrows) and unannotated data (in blue arrows) are used. Unannotated data is converted into "experience" to feed into the experience pool of RL.

module. The NLU module takes the query from the patient as input and parses the user intent and other important information. In our scenario, the user's intent is to request a diagnosis, and other important information includes the patient's symptoms and corresponding attributes. Given the parsed information from NLU, the DM module determines the system's action. After that, the NLG module converts the action into the natural language to respond to the patient. The whole architecture is presented in Fig. 2. In this paper, NLU and NLG modules use the same template-based method in [23], and perform symptom and attribute normalization to eliminate synonyms. The difference lies in the DM module, where we use a few annotated dialogs and massive unannotated dialogs to train the model. We now describe the DM part in detail.

3.2 Dialog Management

Dialog management, the core component of the system, decides how the system acts at each turn. To reduce the demand for annotated data for the fine-grained diagnosis system, we first extract diagnostic logic from unannotated data, then perform an ensemble training guided by the extracted logic in the RL framework. In this section, we first interpret these two steps separately, then illustrate how the unannotated and annotated data flow in the proposed framework.

Diagnostic Logic Extraction. In medicine, diagnostic logic is formally represented as diagnostic decision trees [12]. As shown in Fig. 3, the non-leaf node is the attribute of symptoms k enquired by the doctor, the edge is the value of the attribute v answered by the patient, and the leaf node is the final diagnosis result d. The doctor makes the decision according to the v on every node k until obtaining the result d. In practice, it is hard to build the diagnostic decision tree as it requires extensive medical expertise, but we can extract the "logical paths" of the decision tree from unannotated data, which is much cheaper and more efficient. In Fig. 3, we denote each path from the root to the leaf node as a logical path. In a conversation, the doctor's inquiry about the symptom's attribute corresponds to a node in the decision tree, and the patient's answer corresponds to an edge, so one conversation corresponds to one "logical path" in the tree. For

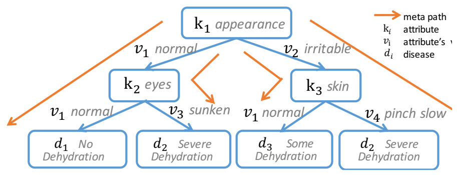


Fig. 3. The diagnostic decision tree presents the diagnosis logic of the doctor. The orange arrows point out the logical paths of diagnosis processes.

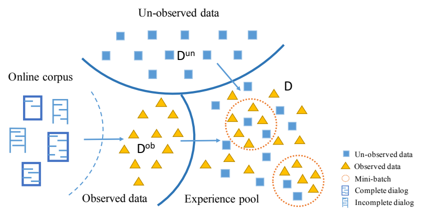


Fig. 4. The experience pool consists of (1) "experience" extracted from unannotated data and (2) simulation records from the system-patient simulator.

the incomplete conversations without the final diagnosis (leaf-node), we treat the path from root to the last doctor's inquiry as a logical path.

Formally, we note the conversation as $c = \{q_1, r_1, \ldots, q_n, r_n\}$ where q_i is the doctor's utterance and r_i is the patient's utterance at the i-th turn. We reuse the $\text{NLU}(\cdot)$ module described in the former section to parse the symptom's attributes k and their value v for each utterance. For each conversation c, we can extract a "logical paths" $p = \{k_1, v_1, \ldots, k_n, v_n\}$ to represent the diagnostic logic.

Ensemble Training. To plug in the diagnostic logic extracted from the unannotated data for training, we employ RL using the experience replay strategy, where the experience pool contains the extracted diagnostic logic from both unannotated and annotated data (see Fig. 4).

Experience replay maintains an "experience pool", in which the "experience" records actions the model took during the exploration. At each training step, The model updates the parameters by minimizing the temporal-difference(TD) error on a mini-batch of "experience" sampled from the pool. The "experience" is formulated as a quadruple $\langle s, a, r, s' \rangle$, where s stands for the state, a stands for the action the model takes under s, r is the instant reward the model receives by taking a, and s' is the next state after taking a.

Since the logical paths obtained from unannotated data indicate how the doctor is supposed to act for diagnosis, we then propose to convert the logical paths into the format that the experience pool takes to substitute the random choices with more reliable ones. In the beginning stage of training, the exploration is almost randomly performed as the model has no medical knowledge about diagnosis, so the training of RL is unstable and may fall into the suboptimal point. We propose to add a warm start stage before training, where we feed the "experience" into the experience pool. In this way, so the initial exploration will be based on real data. Even though this kind of "experience" may not be totally correct, it is at least better than the random one. After the warm start, the model will further improve the policy based on the setting of a relatively reliable parameter, so the model converges more quickly.

Formally, one logical path is a series of symptoms and attributes in the doctor's inquiry order noted as $\{k_1, v_1, k_2, v_2, \cdots, k_n, v_n\}$. We separate former $2 * n - 1$ items as the state s and regard the last item as the action a, thus one path can be converted to one $\langle s, a\rangle$ pair. Then we obtain the instance reward r from the patient's answer in the next turn. For example, in the current turn, the doctor asks whether the patient has taken the vaccine recently. If the answer is yes, it means the doctor has asked about the right symptom that may help the diagnosis, so the reward is positive. If not, it means the doctor may ask about an irrelevant symptom, so the reward is negative. By taking action a, the state updates to s', so all the information can be noted as $\langle s, a, r, s'\rangle$. These quadruple can be inserted into the experience pool to guide the exploration direction.

To further improve the diagnosis efficiency, the symptom the systems ask about are desired to be more "distinguishable" for disease classification. Inspired by the term frequency-inverse document frequency (TF−IDF), which reflects the distinct degree of a word to the document, we propose an action frequency-inverse disease frequency (AF−IDF) to measure whether an action is distinguishable and worth for making diagnosis.

Specifically, we have a disease set $D = \{d_1, \cdots, d_n\}$, and each disease $d_i = \{c_i^1, \cdots, c_i^n\}$ consists of several dialogs c_i^j. For an action a extracted from d_i, AF−IDF is calculated as AF−IDF = AF · IDF, where

$$\mathrm{AF} = \frac{\mathrm{count}((\mathrm{a}, \mathrm{c_i^j})|\mathrm{c_j^i} \in \mathrm{d^i})}{|\{\mathrm{d_i}\}|}, \ \mathrm{IDF} = \log \frac{|\mathrm{D}|}{\mathrm{count}((\mathrm{a}, \mathrm{d_i})|\exists \mathrm{c_i^j} \in \mathrm{d_i}, \mathrm{a} \in \mathrm{c_i^j})}. \quad (1)$$

We note AF−IDF as the confidence score δ of taking action a, and label the experience with it noted as $\langle s, a, r, s', \delta\rangle$. The higher the score δ is, the more distinguishable the action is. For actions from incomplete dialogs without a disease tag, we use the average score over all the diseases $\{d_i\}$ as its confidence score. We normalize the confidence score into (0,1) using the logistic function, multiply the normalized score ${\delta_i}'$ with the instant reward r_i, and then store the experience $\langle s, a, \delta' r, s'\rangle$ into the experience pool noted as D^{ob}. D^{ob} is the observed data as all the experiences are summarized from the real-world conversations.

As mentioned before, we adopt DQN using the experience replay and the target network. Q-learning updated at iteration i uses the objective function as,

$$\mathcal{L}_i(\theta_i) = E_{\langle s,a,r,s'\rangle \sim U(D_i)}[(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2] \quad (2)$$

where $D_i = D_i^{un} \cup D^{ob}$, and $Q(s, a)$ is the function that estimates the maximum of cumulative future reward discounted by γ for taking action a at state s, $U(D_i)$ is the uniform distribution over the experience pool D_i, θ_i are the parameters of the Q-network at iteration i and θ_i^- are the parameters for the target network. The target network parameters θ_i^- are fixed and only updated with the Q-network parameters (θ) every C step. In Fig. 4, D consists of D^{un} from the simulation and D^{ob} from the unannotated data. D^{un} stands for the unobserved data, which is filled with the simulation results. D^{ob} contains the real dialog information that is closer to the real data distribution.

Bidirectional Data Flow. The model uses two kinds of data for training, as shown in the orange and blue arrows in Fig. 2. Annotated data is used as the supervised data to support the patient simulator, which is a straightforward way to guide the direction of gradient descent. Unannotated data is injected into the experience pool to affect the training direction indirectly. By doing so, the model is more likely to explore the space that has already been observed in the human-to-human conversation. The observed data helps the RL-based system to find a reliable warm start instead of a random one, which stabilizes the training process. The annotated data and unannotated data formulate a pair of collaborative learners and benefit the training simultaneously.

4 Experimental Settings

Datasets. We conduct the experiments on fine-grained diseases which are chosen from the *diseases of the circulatory system* category according to ICD-10 [3]. The 8 diseases are: "Acute rheumatic fever", "Hypertensive diseases", "Ischaemic heart diseases", "Pulmonary heart disease and diseases of pulmonary circulation", "Other forms of heart disease", "Cerebrovascular diseases", "Diseases of arteries, arterioles, and capillaries", "Other and unspecified disorders of the circulatory system". We use 245 EHR annotated by qualified doctors as the annotated data and 10k unannotated dialogs from a Chinese inquiring system [4]. The dataset uses 121 different symptoms. Each symptom has 5 attributes, so the action space is 605. We do not use the dataset in [23] since the diseases in it belong to different disease categories in ICD-10, so a few simple symptoms such as diarrhea or cough are enough to distinguish functional dyspepsia and gastroenterology. In contrast, we use a fine-grained disease set coming from the "heart disease" category in ICD-10, which is more challenging and has practical significance.
Hyper-parameters. We use a two-layer multiple layer perception (MLP) DQN with the same hyper-parameters in [23]. The target network and experience replay strategy are used for policy training, as well as the ϵ -greedy exploration strategy where ϵ is set to 0.1. The maximal turn of the conversation is 22.
Competing Methods. DQN. The basic DQN-based method for the training of RL [23]. **DQN-A.** Integrate the action transition probability with DQN for joint action prediction [26]. Here, we use a rule-based method for both NLU and NLG modules to make a fair comparison. **DQN-D.** Integrate the disease-symptom co-occurrence probability with DQN for joint action [26]. The experimental setting is the same as other baselines. **DQN-A-D.** Integrate both the action relation matrix and disease-symptom knowledge with DQN for joint action prediction [26]. **DQN-UN.** The proposed model uses the experience from unannotated data but does not uses AF$-$IDF to make the action more distinguishable. **DQN-UN-D.** The full model that uses both unannotated data and AF$-$IDF.
Evaluation Metrics. • *success.* The success rate of the correct disease prediction indicates the system has made the correct diagnosis within the maximal

[3] https://icd.who.int/browse10/2010/en#/IX
[4] zixun.haodf.com

	Avg				Best			
	success	wrong diagnosis	reward	turn	success	wrong diagnosis	reward	turn
Annotated								
DQN	0.285	0.586	-4.802	6.174	0.331	0.541	-1.826	6.343
Annotated + Knowledge								
DQN-A	0.259	0.664	-5.775	4.796	0.294	0.596	-3.634	5.141
DQN-D	0.247	0.632	-7.131	5.855	0.327	0.635	-1.102	4.368
DQN-A-D	0.256	0.681	-5.759	4.283	0.313	0.681	-1.641	4.283
Annotated + Unannotated								
DQN-UN	0.270	**0.531**	-6.299	7.213	0.305	0.560	-3.505	6.258
DQN-UN-D	**0.319**	0.580	**-2.173**	5.517	**0.361**	**0.518**	**0.358**	5.949

Table 1. The overall performance of all competing methods in three categories. Our methods are listed in the third one. The "Avg" part shows the average results of 25 (5 pieces of training multiply 5 testings) experiments, while the "Best" part presents the best results among 25 experiments. A conversation is counted as "success" only when the system makes the diagnosis within the maximal turn.

turn. • *wrong diagnosis.* The rate of the wrong disease prediction, indicating the system has made the wrong diagnosis within the maximal turn. • *reward.* The average reward over the whole conversation. • *turn.* The average number of conversation turn. Given the fact that RL is inherently unstable, we train all the above competing methods 5 times and run the testing of each model 5 times. All the results listed in the tables are the average results of 25 experiments.

5 Experimental Results and Analyses

The upper bound of diagnosis accuracy is calculated by directly regarding all patient's symptoms as features for a classifier instead of collecting information via interaction. We use Support Vector Machine (SVM) and Logistic Regression (LR) to predict the disease tags. The maximal accuracy is about 0.37 (SVM:0.376, LR:0.370), showing that making the diagnosis in our setting is very challenging. Notice that both SVM and LR directly use all the useful symptoms for classification. According to an article, [10] in JAMA, a top medical journal, the correct diagnosis rate made by human experts in telemedicine is around 0.38, which is similar to our scenario (also focusing on fine-grained diseases). Both [10] and this paper uses online doctor-patient interactions without accessing medical examinations or personal records. Hence, 0.37 is a reasonable value.

• **Overall Performance** Table 1 presents the overall performance on all competing methods, and our two proposed methods achieve the highest success rates among all the competing methods. For the average performance among 25 experiments, DQN provides a borderline. Enhanced by external knowledge, DQN-A adds the action transition information to the policy network, which is expected to achieve better performance. However, DQN-A is worse than DQN in terms of all metrics. The problem may lie in the integrated way of action relation. DQN-A simply sums the action probabilities produced by action relation matrix and the predicted probabilities produced by DQN together, but these two distributions are probably in different scopes. Similarly, DQN-D introduces pre-defined

	1	2	3	4	5	std
DQN	0.285	0.257	0.308	0.331	0.240	0.033
DQN-A-D	0.247	0.234	0.211	0.274	0.313	0.035
DQN-UN-D	0.296	0.288	0.338	0.314	0.361	**0.027**

Table 2. The standard deviation (std) of *success* of 5 times training.

	1	2	3	4	5
DQN*	0.206	0.229	0.275	0.311	0.280
DQN	0.285	0.259	0.308	0.331	0.240
difference	+	+	+	+	-
DQN-UN-D*	0.302	0.301	0.253	0.234	0.373
DQN-UN-D	0.296	0.288	0.338	0.314	0.361
difference	-	-	+	+	-

Table 3. The warm start strategy on different methods. "*" indicates no rule-based warm start applied.

symptom-disease knowledge into the framework. With similar integrated strategy, the results of DQN-D are even worse than DQN-A. The low performance of DQN-A-D is further evidence. Hence, these methods [26] are not robust.

As for our proposed methods, DQN-UN uses the unannotated data but not AF−IDF, and achieves a relatively good performance compared with the knowledge-based method. We also notice that DQN-UN performs worse than DQN, and the conversation turn is longer than other methods. The reason is that the actions that cannot be used to tell different diseases apart waste the interaction turn, so the model gains more negative rewards. DQN-UN-D has the best performance among all methods in terms of all evaluation metrics, indicating that the unannotated data indeed contains useful information as long as we use it in a proper way. In addition, the sum of *success* and *wrong diagnosis* can be regarded as the confidence score of making the diagnosis before the end of the conversation. DQN has 0.879 confidence score to make the diagnosis, while for DQN-UN, the confidence score drops to 0.801. It shows if the exploration direction is effected by the wrong information, the system is less likely to make the diagnosis within the maximal turn. After the making the action more distinguishable, the system is equipped with reliable knowledge, and the confidence score of DQN-UN-D rises to 0.899.

For the best performance among 25 experiments of each method, DQN reaches the *success* of 0.33, and DQN-A is unsurprisingly low. Inconsistent with the average metrics, DQN-D has a better performance. The reason may be that when the predicted action distribution and knowledge-based distribution are in a similar scope, the overall performance will be much more satisfactory. While for DQN-A-D, it is difficult to find a balance between the three items, so the result is not so good. DQN-UN still has a poor performance, but DQN-UN-D gets 0.361 in terms of *success*, which is quite close to the performance of SVM and RL. Reaching almost the upper bound of *success* proves the potential of DQN-based policy network and the effectiveness of unannotated data.

• **Training Stability** RL is inherently unstable, and introducing unannotated data helps to maintain more stable training. We run all the approaches 5 times and record their success rate to estimate their stability as listed in Table 2. The lower the std score is, the lower the deviation between different experiments, and the more stable the approach is. Our model is much more stable than other DQN model. The std value of DQN is 0.033. When it comes to DQN-A-D, the std value arises 6% compared with DQN. In contrast, our full model DQN-UN-D drops 18%, which shows our method alleviates the instability of RL training.

• **Knowledge Summarization** From another perspective, the extraction of "experience" proposed in this paper can be regarded as a knowledge summarization process. In our model, we use the logical paths summarized from unannotated data to formulate the "experience" for the warm start, while conventional DQN designs rules to obtain "experience" for the warm start.

We compare the methods on whether to use the rule-based warm start strategy at the beginning or not. Table 3 shows that DQN benefits from the rule-based warm start in most cases (4/5), while our model DQN-UN-D is not (2/5). We believe it is because the knowledge from the warm start has been already included in the unannotated data, and our model does not rely on the manually-designed rules in the warm start stage. Hence, our model not only reduces the use of annotated data, but also waives the need for designing rules in the warm start.

6 Conclusion

Making the diagnosis of fine-grained diseases is of significance in practical, but it requires more complicated diagnosis logic and training data. To reduce the demand for annotated data in fine-grained diseases diagnosis, we propose an RL-based method to leverage both few annotated dialogs from experts and unannotated dialogs from online forums. The method extracts diagnostic logic from the unannotated dialogs, and build a user simulator using annotated dialogs. The experiments show that our model achieves good performance for 8 fine-grained "heart disease" diagnosis.

7 Acknowledgments

This paper is supported by National Natural Science Foundation of China (NSFC Grant No. 62106275, No.62106008 and No.62006004).

References

1. Adewumi, T., Abid, N., Pahlavan, M., Brnnvall, R., Sabry, S.S., Liwicki, F., Liwicki, M.: Smaaprat: Dialogpt for natural language generation of swedish dialogue by transfer learning (2021)
2. Budzianowski, P., Ultes, S., Su, P., Mrksic, N., Wen, T., Casanueva, I., Rojas-Barahona, L.M., Gasic, M.: Sub-domain modelling for dialogue management with hierarchical reinforcement learning. SIGDIAL pp. 86–92 (2017)
3. Dhingra, B., Li, L., Li, X., Gao, J., Chen, Y., Ahmed, F., Deng, L.: Towards end-to-end reinforcement learning of dialogue agents for information access. ACL **1**, 484–495 (2017)
4. Enayet, A., Sukthankar, G.: A transfer learning approach for dialogue act classification of github issue comments (2020)
5. Glas, N., Prepin, K., Pelachaud, C.: Engagement driven topic selection for an information-giving agent. Workshop on the SPD (2015)
6. Golub, D., Huang, P., He, X., Deng, L.: Two-stage synthesis networks for transfer learning in machine comprehension. EMNLP pp. 835–844 (2017)

7. Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A.: Autotutor: An intelligent tutoring system with mixed-initiative dialogue. IEEE **48**(4), 612–618 (2005)
8. Jaech, A., Heck, L.P., Ostendorf, M.: Domain adaptation of recurrent neural networks for natural language understanding. INTERSPEECH pp. 690–694 (2016)
9. Jovanovic, M., Baez, M., Casati, F.: Chatbots as conversational healthcare services. IEEE Internet Computing **PP**(99), 1–1 (2020)
10. Jr, R.J., M, A., M, S., A, T., A, Y., I, L., CL, K., KE, E.: Choice, transparency, coordination, and quality among direct-to-consumer telemedicine websites and apps treating skin disease. JAMA Dermatol (2016)
11. Kao, H.C., Tang, K.F., Chang, E.Y.: Context-aware symptom checking for disease diagnosis using hierachical reinforcement learning. AAAI (February 2018)
12. Levine, A.C., Glavis-Bloom, J., Modi, P., Nasrin, S., Rege, S., Chu, C., Schmid, C.H., Alamb, N.H.: Empirically derived dehydration scoring and decision tree models for children with diarrhea: Assessment and internal validation in a prospective cohort study in dhaka, bangladesh. Global Health: S & P **3**(3), 405–418 (2015)
13. Li, T., Zhao, Y., Zhang, C., Luo, J., Zhang, X.: A knowledge-guided and data-driven method for building hvac systems fault diagnosis. Building and Environment **198**, 107850 (2021)
14. Luo, H., Li, S.W., Glass, J.: Prototypical q networks for automatic conversational diagnosis and few-shot new disease adaption. arXiv preprint arXiv:2005.11153 (2020)
15. Mo, K., Yang, Q., Fung, P.: Cross-domain dialogue policy transfer via simultaneous speech-act and slot alignment. arXiv preprint arXiv:1804.07691 (2018)
16. Moulya, S., Pragathi, T.R.: Mental health assist and diagnosis conversational interface using logistic regression model for emotion and sentiment analysis. Journal of Physics: Conference Series (1), 012039– (2022)
17. Shah, P., Hakkani-Tür, D., Tür, G., Rastogi, A., Bapna, A., Nayak, N., Heck, L.: Building a conversational agent overnight with dialogue self-play. arXiv preprint arXiv:1801.04871 (2018)
18. Sok, M., Svegl, E., Grabec, I.: A sensory-neural network for medical diagnosis. EAIS pp. 1–6 (2017)
19. Song, Y., Liu, Z., Bi, W., Yan, R., Zhang, M.: Learning to customize model structures for few-shot dialogue generation tasks. ACL pp. 5832–5841 (2020)
20. Tang, K.F., Kao, H.C., Chou, C.N., Chang, E.Y.: Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. NeurIPS (2016)
21. Tran, V., Nguyen, L.: Adversarial domain adaptation for variational neural language generation in dialogue systems. COLING pp. 1205–1217 (2018)
22. Tseng, B., Kreyssig, F., Budzianowski, P., Casanueva, I., Wu, Y., Ultes, S., Gasic, M.: Variational cross-domain natural language generation for spoken dialogue systems. SIGDIAL pp. 338–343 (2018)
23. Wei, Z., Liu, Q., Peng, B., Tou, H., Chen, T., Huang, X., Wong, K.F., Dai, X.: Task-oriented dialogue system for automatic diagnosis. ACL **2**, 201–207 (2018)
24. Williams, J.D.: Web-style ranking and SLU combination for dialog state tracking. SIGDIA pp. 282–291 (2014)
25. Williams, J.D., Zweig, G.: End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:1606.01269 (2016)
26. Xu, L., Zhou, Q., Gong, K., Liang, X., Tang, J., Lin, L.: End-to-end knowledge-routed relational dialogue system for automatic diagnosis. AAAI (2019)
27. Zhao, T., Eskenazi, M.: Zero-shot dialog generation with cross-domain latent actions. SIGDIAL pp. 1–10 (2018)