MTDiag: An Effective Multi-Task Framework for Automatic Diagnosis

Zhenyu Hou[1], Yukuo Cen[1], Ziding Liu[2], Dongxue Wu[2], Baoyan Wang[2], Xuanhe Li[2], Lei Hong[2], Jie Tang[1]

[1]Tsinghua University,
[2]Meituan
{houzy21, cyk20}@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn,
{liuziding, wudongxue03, wangbaoyan, lixuanhe, honglei}@meituan.com

Abstract

Automatic diagnosis systems aim to probe for symptoms (i.e., *symptom checking*) and diagnose disease through multi-turn conversations with patients. Most previous works formulate it as a sequential decision process and use reinforcement learning (RL) to decide whether to inquire about symptoms or make a diagnosis. However, these RL-based methods heavily rely on the elaborate reward function and usually suffer from an unstable training process and low data efficiency. In this work, we propose an effective ***m***ulti-***t***ask framework for automatic ***diag***nosis called MTDiag. We first reformulate symptom checking as a multi-label classification task by direct supervision. Each medical dialogue is equivalently converted into multiple samples for classification, which can also help alleviate data scarcity problem. Furthermore, we design a multi-task learning strategy to guide the symptom checking procedure with disease information and further utilize contrastive learning to better distinguish symptoms between diseases. Extensive experimental results show that our method achieves state-of-the-art performance on four public datasets with 1.7%∼ 3.1% improvement in disease diagnosis, demonstrating the superiority of the proposed method. Additionally, our model is now deployed in an online medical consultant system as an assistant tool for real-life doctors.

Introduction

Artificial intelligence is revolutionizing our life in various aspects and has the potential to bring new vitality to the healthcare and medical domain. Automatic diagnosis (Li et al. 2017; Wei et al. 2018; Xu et al. 2019), which aims to provide convenient medical care and assist diagnosis, is one of the most promising applications. The rapidly growing and aging population brings an increasingly heavy workload for read-life doctors, especially in countries and areas with high-density populations. And in the Internet era, people are also seeking more convenient ways to find medical services during the COVID-19 pandemic. Thus automatic diagnosis arises at this moment and is gaining increasing attention in contemporary research. Currently, the main focus has been on making more effective diagnostic decisions or building a diagnostic dialogue system (Shivade et al. 2014; Xia et al. 2020; Chen et al. 2022).

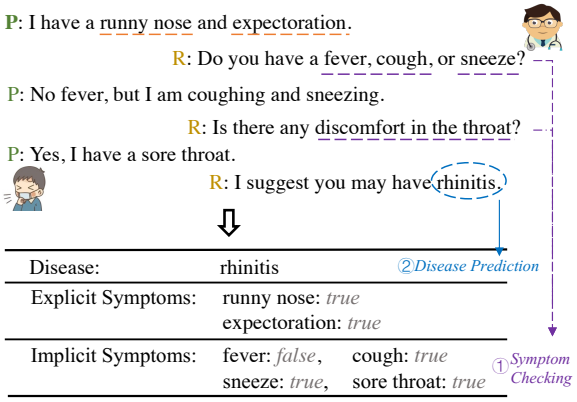

Figure 1: A medical dialogue can be converted to a standard *user goal* for automatic diagnosis, including a disease tag, explicit symptoms, and implicit symptoms.

Specifically, the automatic diagnosis task usually relies on interactions between an agent and a patient, where the agent collects necessary symptoms for the diagnosis. This is consistent with the real-world diagnostic procedure. As the example in Figure 1 shows, the patient first states a self-report. Then the doctor checks several related symptoms and finally gives a diagnostic suggestion to the patient. The medical dialogue can be simplified and converted to a corresponding diagnosis sample (or user goal), consisting of explicit symptoms obtained from the user's self-report, additional implicit symptoms from inquiries, and a disease tag to be predicted. Hence, the problem can be viewed as a multi-step reasoning task (Chen et al. 2022) and targets inquiring about the implicit symptoms step by step based on explicit symptoms and then making the final disease diagnosis. Note that *in automatic diagnosis, the agent only asks about symptoms, and the patient answers with Yes/No/Not sure, which is quite different from the natural language used in traditional task-oriented dialogue systems.*

Previous works (Wei et al. 2018; Xu et al. 2019; Liao et al. 2020) for automatic diagnosis typically regard the problem as a Markov Decision Process (MDP) (Young et al. 2013) and address it via reinforcement learning (RL) (Cuayáhuitl, Keizer, and Lemon 2015; Yu et al. 2021). For example,

the dialogue policy can be parameterized with a deep Q-network (Mnih et al. 2015; Hessel et al. 2018). However, RL-based methods suffer from potential drawbacks, especially in the medical domain. On the one hand, RL needs explicit learning objectives and elaborate rewards, making it hard to balance symptom checking and disease diagnosis. Learning the action merely from the final reward is not only less data-efficient but also inconsistent with the actual diagnostic procedure, where doctors would adjust inquiries based on instant response. In addition, RL is data-hungry and usually requires a considerable amount of data to achieve satisfactory results. Unfortunately, the data is always sparse and insufficient in the medical domain. Recent effort (Chen et al. 2022) considers automatic diagnosis as a sequence generation task and generates implicit symptoms in an auto-regressive style. Nevertheless, as the symptoms are intrinsically unordered, it is necessary to preserve this inductive bias in the algorithm design.

In this work, we propose an effective multi-task framework, MTDiag, to address these challenges. We first reformulate the symptom checking as a multi-label classification task to keep the unordered setting. A multi-turn dialogue can be transformed into a set of $(\text{input}, \text{label})$ samples, where the inputs represent the known symptoms, and the labels are the symptoms to be inquired. To be specific, a dialogue consisting of k implicit symptoms can be decomposed into $\sum_{i=0}^{k} \binom{k}{i}$ samples. Based on this decomposition, we can transfer the sequential decision process within one multi-turn dialogue into multiple independent training samples of the multi-label classification task. Secondly, we propose an effective multi-task learning strategy to better capture the relationship between disease and symptom. The intuition is that in real diagnosis, when checking possible symptoms, doctors use a combination of their prior experiences of co-occurring symptoms and their professional knowledge of what disease might cause the symptoms. To leverage this prior knowledge, we employ two task-specific attentional pooling heads for predicting target symptoms and disease based on a Transformer (Vaswani et al. 2017) encoder. As contrastive learning can push samples to form better clusters, we also use contrastive learning to differentiate symptoms of different diseases. Our model has been deployed online, serving hundreds of thousands of people every day.

Our main contributions are summarized in the following:

- We reformulate symptom checking as a multi-label classification task while keeping the unordered nature of automatic diagnosis. Our approach could alleviate data scarcity in the medical field and speed up training.
- We design a multi-task learning framework to interweave the learning of symptom and disease prediction. Specially, we employ contrastive learning to better distinguish symptoms among different diseases.
- Extensive experimental results show that the proposed method achieves state-of-the-art performance on four public medical diagnosis datasets, demonstrating the effectiveness of our approach.

input	label
es_1, es_2	[-, -, 1, 1, 0, 0]
es_1, es_2, is_1	[-, -, -, 1, 0, 0]
es_1, es_2, is_2	[-, -, 1, -, 0, 0]
es_1, es_2, is_1, is_2	[-, -, -, -, 0, 0]

Table 1: A simple example of decomposing a user goal "$E = (es_1, es_2) = (s_1, s_2), I = (is_1, is_2) = (s_3, s_4)$" of multi-turn dialogue into multi-label classification, assuming there are only 6 symptoms in total. The 4 pieces could cover all the information contained in the user goal. The "-" in the label represents masked symptoms appearing in the input.

MTDiag Framework

In this section, we first introduce how to reformulate symptom checking into a multi-label classification task. Then we propose a simple and effective attention-based model and a multi-task learning strategy to tackle the problem of both symptom checking and disease diagnosis.

Problem Reformulation

Formally, a sample of automatic diagnosis data contains explicit symptoms $S_{ex} = \{es_1, ..., es_n\}$, implicit symptoms $S_{im} = \{is_1, ..., is_m\}$, and a disease tag *Dis*. Only the explicit symptoms are accessible at the beginning. The target of symptom checking is to obtain as many implicit symptoms as possible via limited turns of inquiries since more implicit symptoms would contribute to a more precise diagnosis. For each symptom inquiry, the simulator will output *True* or *False* as an answer for a positive/negative symptom and *not sure* for symptom not in the user goal $S_{ex} \cup S_{im}$. The objective equals maximizing the likelihood $P(S_{im}|S_{ex})$. We denote the symptoms obtained via inquires as $S_{add} \subseteq S_{im}$, and the missed ones as $\bar{S}_{add} = S_{im} - S_{add}$. Since symptoms are naturally orderless, the learning objective can be formulated as follows:

$$\prod_{S_{add} \subseteq S_{im}} P(\bar{S}_{add}|S_{ex} \cup S_{add}) \tag{1}$$

We model the symptom checking as a multi-label rather than multi-class classification task to avoid the potential problem of sequential generation. Afterward, the disease is predicted based on known symptoms, whose learning objective is to maximize $P(Dis|S_{ex} \cup S_{add})$.

Training. In this part, we show how to apply supervised learning to tackle the problem of multi-step reasoning. Traditional supervised learning hypothesizes that data samples are independent. We aim to decompose a multi-turn diagnostic dialogue into several independent one-step multi-label classification data samples while covering all possible cases and information in the dialogue. According to Equation 1, to maximize $p(S_{im}|S_{ex})$, we could maximize each $P(\bar{S}_{add}|S_{ex} \cup S_{add})$ independently, which corresponds to an intermediate state before an inquiry of a dialogue: given explicit symptoms S_{ex} and observed implicit ones S_{add} as input, the objective is to predict the remaining implicit symptoms $\bar{S}_{add}$. In such a case, the problem can be converted to

a multi-label classification task:

$$\text{Input}(S_{ex} \cup S_{add}) \xrightarrow{predict} \text{Label}(\bar{S}_{add})$$

If we enumerate all possible S_{add} of each dialogue, that is, all subsets of S_{im}, any intermediate state would be covered during training, and we transfer the sequential decision problem into a multi-label classification task under the setting of supervised learning. Table 1 shows a simple example of the decomposition of a user goal containing two explicit and two implicit symptoms. Symptoms in the input should not appear in the label in order to prevent label leakage and false negatives. In our implementation, we mask the input symptoms label during training.

This decomposition also has advantages. A dialogue with k implicit items can be transformed into $\sum_{i=0}^{k} \binom{k}{i}$ training samples at most, which significantly increases the scale of training data and helps alleviate the problem of data scarcity. This technique makes more sense in the medical domain because it is difficult and costly to collect real data. Beyond this, mini-batch training under a supervised setting runs and converges much faster than reinforcement learning.

Inference. In the inference, we still follow the multi-turn setting to imitate the actual medical dialogue scenario. In each turn, the model accepts $S_{ex} \cup S_{add}$ as inputs, and the symptom of the highest probability in the prediction is selected as the subsequent inquiry. If the patient answers *True* or *False*, the symptom will be marked and added to the known set. Otherwise, the next highest probability symptom will be the subsequent inquiry until finding the implicit one or stopping. To make the model aware of when to stop, we set a *stop threshold* $\delta \in (0, 1)$ as the minimum probability boundary. If the probability of all remaining symptoms in the prediction is all below δ, the model will stop. Then the explicit symptoms and those obtained via inquiries will be used for disease diagnosis.

Model

In this part, we introduce our proposed attention-based model and a multi-task learning strategy to resolve symptom checking and disease diagnosis. The architecture is illustrated in Figure 2.

Model Architecture. In each step, our model maps an input set of known symptoms $(s_1, ..., s_n)$ to a set of continuous representations and then aggregates them together to make the prediction. All symptoms are converted to d dimension token embeddings, denoted as $\boldsymbol{x}_i \in \mathbb{R}^d$ for symptom s_i. Explicit and implicit symptoms share the same token embeddings. We add symptom condition embedding $\boldsymbol{c}_i \in \mathbb{R}^d$ to indicate it is positive (*True*) or negative (*False*), which works similarly to the positional encoding in Transformer (Vaswani et al. 2017).

We first stack multiple Transformer blocks to capture the interaction between symptoms, as various works have demonstrated that transformer is powerful in tackling sequences of varying lengths. To be concrete, after adding the condition embeddings, we feed the symptom embeddings to the transformer encoder to get hidden representations:

$$[\boldsymbol{h}_1, .., \boldsymbol{h}_n] = \text{MH-Attn}(f_Q(\tilde{\boldsymbol{X}}), f_K(\tilde{\boldsymbol{X}}), f_V(\tilde{\boldsymbol{X}})),$$

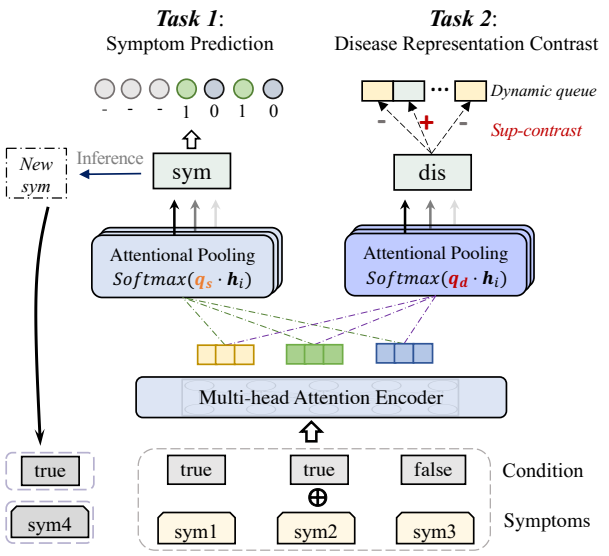

Figure 2: The multi-task learning framework. Our model first employs a transformer to encode input symptoms and their conditions. Then we predict the next possible symptom and disease representation using distinct attentional pooling heads, respectively.

where $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, .., \tilde{\boldsymbol{x}}_n], \tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{c}_i$, and $f(\tilde{\boldsymbol{X}})$ denotes the transformation for query, key, and value. MH-Attn simply represents stacked multi-head attention block.

Previous studies (Reimers and Gurevych 2019) show that inserting a special [CLS] token in the sequence often achieves good performance for sentence-level classification. In our experiments, we find that aggregating symptoms' representation works better in symptom checking. In this work, we design a simple and effective attentional pooling to obtain the sequence representation for the final prediction.

We first construct a virtual *signal* using a shared learnable vector $\boldsymbol{q} \in \mathbb{R}^d$ to represent the target disease/symptom. The signal is employed to calculate the attention scores for aggregation. This works similarly to the self-attention used in Graph Attention Networks (Veličković et al. 2018), and the main difference is that our query is a learnable vector rather than any input representation.

$$a_i = \frac{\exp(\phi(\boldsymbol{q} \cdot \boldsymbol{h}_i / \alpha))}{\sum_{j \in \mathcal{N}} \exp(\phi(\boldsymbol{q} \cdot \boldsymbol{h}_j)/\alpha)}, \tag{2}$$

where $\mathcal{N}$ is the set of symptoms in the input set, $\phi(x)$ is the LeakyReLU activation and, $\alpha \in \mathbb{R}^+$ is the temperature. We obtain the prediction by computing the linear combination of symptom embeddings and also utilize multi-head attention to improve the expressiveness:

$$\boldsymbol{z} = \boldsymbol{W}_2(\Big\|_{k=1}^{K} \sigma(\sum_{i \in \mathcal{N}} a_i^{(l)} \boldsymbol{h}_i)). \tag{3}$$

Multi-task Design. From the above description, we aim to predict the implicit symptoms based on explicit ones. The

multi-label training objective tends to guide the model to learn the concurrence of symptoms. Despite the fact that we always predict diseases based on symptoms in our daily life, it is the disease itself that causes symptoms to appear. Thus this inspires us to provide auxiliary information about the disease to support symptom checking. As illustrated in Figure 2, we design a multi-task learning strategy to achieve this goal. In symptom checking, we employ two different attentional pooling heads for symptom and disease prediction respectively. A transformer serves as a shared bottom encoder of the two heads to capture information of two tasks. For clarity, we denote the output of the symptom prediction head as $\boldsymbol{z}^{(s)} \in \mathbb{R}^C$ and the disease head $\boldsymbol{z}^{(d)} \in \mathbb{R}^d$, where C is the number of symptoms.

For symptom prediction, binary cross entropy (BCE) is a traditional solution to multi-label classification for training. Peng et al. (2020) point out that BCE declines the suppression between categories and behaves poorly in imbalanced multi-label distribution. In automatic diagnosis, there could be hundreds of symptoms in total, but each user goal usually involves only less than ten symptoms. To tackle the issue, we use the concurrent softmax proposed in (Peng et al. 2020):

$$\mathcal{L}_{sym} = -\sum_{i=1}^{C} y_i \log \frac{\exp(z_i^{(s)})}{\sum_{j=1}^{C}(1-y_j)\exp(z_j^{(s)}) + \exp(z_i^{(s)})}$$

For the auxiliary disease prediction, we resort to the idea of contrastive learning (He et al. 2020). Contrastive learning implicitly pulls clusters of points belonging to the same class together while pushing apart samples from different classes. In our scenario, it agrees with the aim of symptom checking to pull the combination of symptoms belonging to the same disease together and separate irrelevant ones. Supervised contrastive learning (Khosla et al. 2020) generalizes self-supervised contrastive learning to an arbitrary number of positive samples. Samples belonging to the same disease are all viewed as positive:

$$\mathcal{L}_{aux} = \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\boldsymbol{z}^{(d)} \cdot \boldsymbol{z}_p^{(d)}/\tau)}{\sum\limits_{a \in N(i)} \exp(\boldsymbol{z}^{(d)} \cdot \boldsymbol{z}_a^{(d)}/\tau)}.$$

Here, $P(i)$ includes all positive samples of $\boldsymbol{z}^{(d)}$, $N(i)$ is the set of negative samples, and $\tau \in \mathbb{R}^+$ is a temperature hyperparameter. As demonstrated in (Chen et al. 2020; He et al. 2020), contrastive learning benefits from larger batch size and more negative samples. However, automatic diagnosis suffers from insufficient training data and a small batch size. To alleviate this issue, we adopt a *dynamic queue*, which stores representations of previous samples to serve as positives/negatives. As the training continues, we progressively update the queue by adding the latest samples and removing the oldest ones. This enables us to use large negative samples with limited batch size. Note that in our method, $S_{ex} \cup S_{add}$ is used for symptom prediction while $S_{ex} \cup S_{im}$ for disease prediction. We find that predicting the disease with only partial symptoms would potentially bring extra noise and harm the diagnosis performance.

Finally, we add the two losses together for the training of symptom checking with a coefficient λ:

$$\mathcal{L} = \mathcal{L}_{sym} + \lambda \cdot \mathcal{L}_{aux}.$$

Then the attention head for disease prediction can be directly used for disease prediction. In our experiments, we find that training a new attentional head without transformer encoders from scratch achieves better performance.

Experiments

In this section, we conduct extensive experiments on public datasets of automatic diagnosis to evaluate our method.

Setup

Datasets. We evaluate our method on four commonly used public datasets. The MDD dataset is from ICLR 2021 Workshop MLPCP Track 2 Medical Dialogue System for Automatic Diagnosis[1]. It contains 2,374 user goals and 118 symptoms, covering 12 disease types. All the dialogues in MDD are derived from real-world patients in offline hospitals, thus closer to the real clinical diagnosis scenario. Since the test set of the MDD dataset is not available, we only report the metric in the validation (dev) set. The MZ dataset (Wei et al. 2018) is collected from the pediatric department in a Chinese online healthcare community (Baidu Muzhi). It contains 710 user goals and 66 symptoms, covering 4 types of diseases. The DXY dataset (Xu et al. 2019) is collected from a Chinese online healthcare community (dxy.com) where users ask doctors for medical diagnoses or professional medical advice. The dataset contains 527 user goals and 41 symptoms, covering 5 types of diseases. The Synthetic dataset (Liao et al. 2020) is constructed from a symptom-disease database called SymCat[2]. It contains 30,000 user goals with 90 diseases.

Baselines. We compare our model with several baselines, including traditional methods and state-of-the-art methods. **SVM** (Chang and Lin 2011) is a commonly used traditional classifier. "SVM-ex&im" can be viewed as a strong baseline as it uses all explicit and implicit symptoms. RL-based methods formulate the medical dialogue as a Markov decision process with reinforcement learning. The **Basic DQN** is from (Wei et al. 2018) and the **PPO** baseline is provided by (Teixeira, Maran, and Dragoni 2021). **HRL** (Liao et al. 2020) integrates a two-level hierarchical policy learning strategy. **KR-DS** (Xu et al. 2019) is an extension of Basic-DQN and integrates relation encoding to help symptom checking and a knowledge-routed graph branch for action decision-making. It also makes use of the self-report of patients before the dialogue. **GAMP** (Xia et al. 2020) integrates the Generative Adversarial Network into the reinforcement learning model with policy gradient and uses mutual information to further enhance the reward function. **Diaformer** (Chen et al. 2022) formulates the dialogue-based diagnosis system as a sequence generation task and designs a transformer-based framework for automatic diagnosis.

[1] https://competitions.codalab.org/competitions/29706

[2] www.symcat.com

	MDD			MZ			DXY			Synthetic		
	Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn
SVM-ex	70.3	-	-	59.0	-	-	64.4	-	-	34.1	-	-
SVM-ex&im	84.5	-	-	71.0	-	-	77.9	-	-	73.2	-	-
Basic DQN	46.4	-	-	65.0	30.1	**3.1**	73.1	32.2	2.9	35.6	2.0	**2.0**
HRL	-	-	-	69.4	27.6	3.5	69.5	16.1	**2.4**	49.6	33.8	8.4
KR-DS	-	-	-	73.0	-	3.4	74.0	-	3.4	-	-	-
GAMP	-	-	-	73.0	-	6.3	76.9	-	3.3	-	-	-
PPO	-	-	-	73.2	-	6.3	74.6	-	3.3	61.8	-	12.6
Diaformer	86.0	87.4	18.9	74.2	75.2	15.3	82.9	82.7	13.1	73.3	90.6	13.7
MTDiag	**89.1**	**89.2**	**13.8**	**75.9**	**79.4**	17.9	**85.4**	**91.3**	12.5	**75.4**	**90.7**	15.1

We report results from previous works if available. Otherwise, if code is provided and could be run successfully, we implement them based on the official code and report the results.

Table 2: Experimental results of four datasets in disease diagnosis. "Acc" is the accuracy of diagnosis. "Recall" is the recall of implicit symptom for symptom checking, and "ATurn" is average turn of inquiry.

Evaluation Metrics. Following the setting of the previous works (Wei et al. 2018; Xu et al. 2019; Chen et al. 2022), we evaluate our method by three metrics: accuracy for disease diagnosis, recall of implicit symptoms, and average inquiry turns for symptom checking. The accuracy is the key metric for automatic diagnosis. The recall and average turn could evaluate the efficiency of the inquiry.

Implementation Details. We implement our model by PyTorch and train the model on NVIDIA 2080Ti (11G). We repeat the experiments five times with random initialization and report the mean results. For symptom checking, the maximum number of turns is set to 20. In addition, in training the diagnosis classifier, explicit symptoms with implicit symptoms in the input and the ones from symptom checking are viewed as different training samples. This could be viewed as a data augmentation technique to help enrich the training data and relieve the problem of data scarcity.

Main Results

Overall Performance. We report results of baselines from previous works if available. For those results that are not previously reported, we run the official code if it is publicly available. All experimental results are shown in Table **??**. Overall speaking, we observe that our approach achieves state-of-the-art or competitive results on both symptom checking and disease diagnosis in the four datasets. MTDiag significantly outperforms reinforcement learning (RL) based methods, especially in DXY and Synthetic datasets where the absolute improvement is at least 10.8% and 13.9% in diagnosis accuracy. For the non-RL-based method Diaformer, our method also has an advantage with an average improvement of 1.9% in the four datasets. For symptom checking, MTDiag tends to request more inquiry runs to achieve a higher recall of symptoms. This is practical and reasonable in real scenarios because more symptoms would help the doctor make a more accurate diagnosis. Compared with Diaformer, MTDiag achieves higher recall and diagnosis accuracy while consuming fewer turns in MDD and DXY. This indicates that our method has more potential to provide valid and informative inquiries of symptoms for diagnosis. Overall, these results demonstrate the effectiveness of the proposed learning framework.

Effect of maximum limited turns. As shown in Table **??**, our method, together with Diaformer, tends to request more inquiry turns than RL-based methods to achieve a higher recall of symptoms. We conduct experiments with 5/10/15 maximum turns to test the performance within fewer turns. Due to the limitation that KR-DS, GAMP, and PPO have not released their code, we compare with two RL baselines (DQN and HRL) and one sequence-generation-based model Diaformer. The results are in Table **??**. It is observed that in most cases, MTDiag can outperform baselines in terms of diagnostic accuracy and recall of implicit symptoms. Specifically, in the setting of 5 limited turns, our method still has a distinct advantage over two RL-based methods with at least 5% improvement in accuracy and 6% in recall on average. MTDiag shows an edge over Diaformer, especially when limited turns are set to 10/15. These results manifest that the proposed approach can achieve satisfactory performance within limited turns.

Effect of stop threshold δ. In the inference of symptom checking, we employ a threshold δ to control when to stop inquiring. In addition to limiting the maximum turns, adjusting the stop threshold is another way to control the inquiry turns. We explore the effect of δ in the MDD dataset, and the results are illustrated in Figure 3. We observe that as the stop threshold δ increases, the recall of implicit symptoms and diagnosis accuracy decrease. This is in line with our intuition that higher δ would cause the inquiry to end earlier, and more implicit symptoms would be overlooked. Besides, it indicates that higher recall can lead to a more accurate diagnosis. Note that the recall drop is more significant than diagnosis accuracy, which may imply that our method can inquire about key implicit symptoms in early steps. Overall speaking, this provides another way to balance the average turn and effectiveness of diagnosis.

Turn	Model	MDD			MZ			DXY			Synthetic		
		Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn
5	Basic DQN	-	-	-	64.1	29.2	2.9	64.7	31.1	2.5	35.6	2.0	**2.0**
	HRL	-	-	-	67.6	26.5	**2.8**	70.2	15.2	**1.9**	44.3	2.4	4.3
	Diaformer	**85.3**	58.3	**4.9**	72.2	**47.2**	5.0	**76.6**	54.5	4.8	49.4	**46.1**	4.9
	MTDiag	82.8	**59.5**	5.0	**72.6**	45.3	5.0	76.1	**58.1**	5.0	**51.1**	44.1	5.0
10	Basic DQN	-	-	-	68.3	29.6	3.0	71.5	32.2	2.7	35.6	2.0	**2.0**
	HRL	-	-	-	69.7	26.6	**3.3**	71.8	15.9	**2.3**	48.8	30.7	7.4
	Diaformer	84.9	75.6	**9.0**	73.1	**65.5**	9.8	80.6	77.8	9.6	63.2	**73.6**	9.6
	MTDiag	**85.9**	**80.1**	9.6	**74.6**	63.2	10.0	**81.9**	**82.7**	9.6	**63.6**	72.5	10.0
15	Basic DQN	-	-	-	68.3	29.7	**3.0**	71.2	32.0	2.7	35.6	2.0	**2.0**
	HRL	-	-	-	70.2	27.2	3.4	71.8	15.9	**2.3**	49.9	32.2	8.3
	Diaformer	85.7	81.8	**12.3**	74.2	73.1	13.8	82.8	82.6	12.4	71.1	86.6	12.6
	MTDiag	**87.5**	**87.2**	12.9	**74.6**	**73.5**	15.0	**85.4**	**89.8**	11.9	**73.3**	**87.9**	14.0

Table 3: Results with smaller different limited turns.

	MDD			MZ			DXY			Synthetic		
	Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn	Acc	Recall	ATurn
MTDiag	**89.1**	**89.2**	14.4	**75.9**	79.4	17.9	**85.4**	**91.3**	12.5	75.4	90.7	15.1
w/o SupCon	88.1	87.3	**12.8**	74.2	**80.0**	18.0	84.0	88.9	12.6	**75.6**	**91.3**	15.1
w/ BCE only	88.2	87.9	13.4	73.8	62.6	18.0	84.1	87.7	**12.1**	74.0	89.5	**14.0**
w/ BCE+SupCon	88.7	87.6	15.9	74.2	62.8	**17.1**	84.8	90.3	12.5	74.1	89.2	14.9

Table 4: Ablation study of different training variants. "w/o SupCon" represents training with symptom prediction loss only. "w/ BCE" means replacing concurrent-softmax based loss with binary cross entropy (BCE) loss.

Ablation Study. We conduct a series of ablation studies to verify the effect of each component in our approach. In this work, we introduce a multi-task learning strategy to assist the symptom checking with extra disease information. Table **??** indicates that this strategy generally boosts the performance of symptom checking as both the recall of implicit symptoms and diagnosis accuracy increase in MDD, MZ, and DXY with almost equal average turns. The Synthetic dataset is an exception in which the performance almost keeps. One possible assumption is that as its scale is much larger than the others, the model is capable of learning well only based on the concurrence of symptoms. We also conduct experiments using binary-cross-entropy (BCE) as the loss function, which is widely used in multi-label classification tasks. The results in Table **??** show that both concurrent softmax (CCE) and BCE perform equally well without disease information. Under the multi-task learning setting, CCE has a slight edge over BCE. These results indicate that the multi-task learning strategy help to improve the performance of automatic diagnosis.

Case Study. We give an actual example from the MZ dataset to demonstrate how symptom checking helps the final diagnosis, as illustrated in Table **??**. Our model first gives an initial but wrong prediction (i.e., Pediatric Diarrhea) based on explicit symptoms before the checking and then inquires about five symptoms step by step. During the 5-turn inquiries, two implicit symptoms (i.e., *loose stool* and *vomiting*) are found in Turns 2 and 3. When the conditions of two key symptoms are recognized, our model gives the right disease prediction, *Pediatric Dyspepsia*. Finally, our model retains the correct diagnosis after inquiring about two relevant but unknown symptoms. Although some implicit symptoms are not found within five inquiries, our method still gives the right prediction as the final diagnosis.

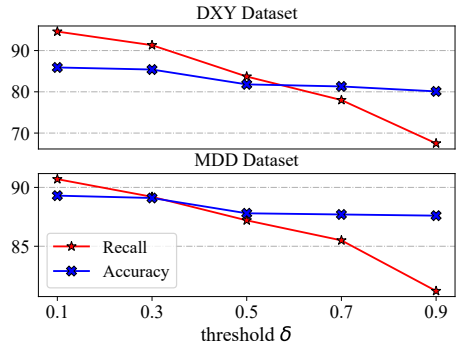

Figure 3: Sensitivity of the stop threshold δ in symptom checking in MDD and DXY datasets. With the increase of δ, symptom recall drops significantly, and diagnosis accuracy decreases slightly due to the fewer symptoms.

Online Deployment. Our method is now deployed as an

Disease tag: Pediatric Dyspepsia **Explicit symptoms**: {green stool: True, diarrhea: True} **Implicit symptoms**:{loose stool: True, vomiting: True, flatus: True}	

Turns	Symptom Inquiry	Disease Prediction
Initial	-	Pediatric Diarrhea
Turn 1	(fever, UNK)	Pediatric Diarrhea
Turn 2	(loose stool, True)	Pediatric Diarrhea
Turn 3	(vomiting, False)	Pediatric Dyspepsia
Turn 4	(watery stool, UNK)	Pediatric Dyspepsia
Turn 5	(runny nose, UNK)	Pediatric Dyspepsia
Diagnosis		Pediatric Dyspepsia

Table 5: Case study of an example chosen from the MZ dataset with 5-turn inquiries. We report the symptom, its condition, and the disease prediction in each turn, where *UNK* means the condition of the symptom is unknown. After the symptom checking, our model makes the correct diagnosis, Pediatric Dyspepsia.

important component of an online medical consultant system as an assistant tool for the real-life doctor, serving hundreds of thousands of users every day. Practically, for each dialogue, we first extract initial symptoms from the user's self-report by named entity recognition and linking tools from the user's self-report. Then our model is activated to perform the multi-turn dialogue to collect implicit symptoms. In each turn, our model predicts the top-k most probable symptoms, which are present as a multiple-selection checkbox. The user could select several symptoms he/she has, and our method will generate subsequent symptoms. Some hand-crafted rules are combined with our method to collect basic information like age and duration of symptoms and avoid any possible offensive inquiries. After a few interactions, all collected symptoms and the diagnostic suggestion of the disease are provided to the doctor for reference. And the doctor could adopt the model's suggestion or inquire for more details to help make the final diagnosis.

Related Work

Previous works mostly view automatic diagnosis as a sequential decision problem. Tang et al. (2016) formulate automatic diagnosis as symptom checking and disease diagnosis and firstly adopts reinforcement learning (RL) to tackle the problem. Kao, Tang, and Chang (2018) introduce medical context into symptom checking and employ a hierarchical RL approach to make a joint diagnostic decision. Xu et al. (2019) incorporate rich prior medical knowledge through a knowledge graph to guide policy learning. Liao et al. (2020) classify diseases into distinct groups according to symptom distribution and builds a hierarchical RL framework to handle inevitably large action space. Xia et al. (2020) borrow the ideas of generative adversarial networks (Goodfellow et al. 2014) and mutual information to improve reward function, thus significantly boosting the performance. Yu et al. (2021) systematically review the development and applications of reinforcement learning in automated medical diagnosis. However, RL-based methods generally suffer from low data efficiency, and it is difficult to achieve satisfying results with very limited data in the medical domain.

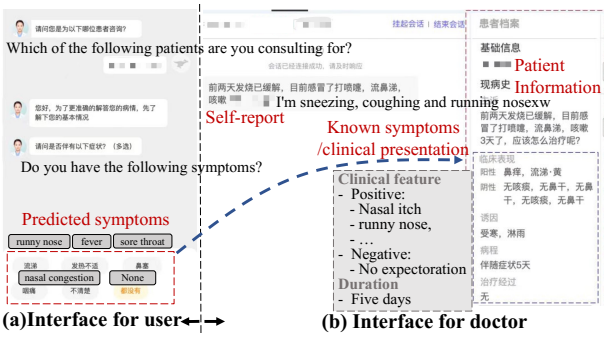

Figure 4: The deployed online medical consultation service. Our method is used to check symptoms of patients before they communicate with the doctor.

Recently, another line of work (Chen et al. 2022) formulates automatic diagnosis as a sequence generation problem. It generates an implicit symptom sequence conditioned on the patient's self-report under the auto-regressive framework. Although it attempts to alleviate the bias resulting from the discrepancy between the ordered generation and the intrinsic disorder of golden implicit symptoms via several orderless techniques, it is fundamentally plagued by the challenge of learning specific order of symptoms.

Conclusion and Future Work

In this paper, we propose MTDiag, an effective multi-task framework for automatic medical diagnosis. We reformulate the symptom checking under a multi-label classification setting and further design a multi-task strategy to guide the training with disease information. MTDiag achieves state-of-the-art performance on four public datasets, demonstrating the effectiveness of our method. As for future work, we identify the importance of high-quality datasets, since they play a significant role in advancing a research field. But current datasets of automatic medical diagnosis are either too small or not fully open-source. This greatly hinders the development of automatic diagnosis. In the future, we will try to build a better benchmark to achieve a more reliable evaluation of existing methods.

Ethical Statement. Artificial intelligence can assist people in a variety of patient care and intelligent health systems. Automatic diagnosis is an important application that helps patients with self-diagnosis or doctors as auxiliary tools. Although our approach achieves promising results, the predicting errors caused by the inadequate data may bring potential harm to users when directly applying the method as a diagnostic system. Under the ethical considerations, our model is deployed as an auxiliary tool to offer suggestions and help doctors check symptoms and make the diagnosis in online medical consultation, rather than serve the patient directly and independently.

Acknowledgements

This work is supported by The National Key Research and Development Program of China (2021YFF1201300), National Science Foundation for Distinguished Young Scholars (No. 61825602), and Meituan.

References

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *TIST*, 2(3): 1–27.

Chen, J.; Li, D.; Chen, Q.; Zhou, W.; and Liu, X. 2022. Diaformer: Automatic Diagnosis via Symptoms Sequence Generation. In *AAAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607.

Cuayáhuitl, H.; Keizer, S.; and Lemon, O. 2015. Strategic dialogue management via deep reinforcement learning. *arXiv preprint arXiv:1511.08099*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative Adversarial Nets. In *NeurIPS'14*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*.

Kao, H.-C.; Tang, K.-F.; and Chang, E. 2018. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *AAAI'18*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*.

Li, X.; Chen, Y.-N.; Li, L.; Gao, J.; and Celikyilmaz, A. 2017. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*.

Liao, K.; Liu, Q.; Wei, Z.; Peng, B.; Chen, Q.; Sun, W.; and Huang, X. 2020. Task-oriented Dialogue System for Automatic Disease Diagnosis via Hierarchical Reinforcement Learning. *arXiv preprint arXiv:2004.14254*.

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*, 529–533.

Peng, J.; Bu, X.; Sun, M.; Zhang, Z.; Tan, T.; and Yan, J. 2020. Large-scale object detection in the wild from imbalanced multi-labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9709–9718.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

Shivade, C.; Raghavan, P.; Fosler-Lussier, E.; Embi, P. J.; Elhadad, N.; Johnson, S. B.; and Lai, A. M. 2014. A review of approaches to identifying patient phenotype cohorts using electronic health records. *Journal of the American Medical Informatics Association*, 21(2): 221–230.

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 *NeurIPS Workshop on Deep Reinforcement Learning*.

Teixeira, M. S.; Maran, V.; and Dragoni, M. 2021. The interplay of a conversational ontology and AI planning for health dialogue management. In *Proceedings of the 36th annual ACM symposium on applied computing*, 611–619.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wei, Z.; Liu, Q.; Peng, B.; Tou, H.; Chen, T.; Huang, X.-J.; Wong, K.-F.; and Dai, X. 2018. Task-oriented dialogue system for automatic diagnosis. In *ACL'18 (Volume 2: Short Papers)*.

Xia, Y.; Zhou, J.; Shi, Z.; Lu, C.; and Huang, H. 2020. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In *AAAI'20*.

Xu, L.; Zhou, Q.; Gong, K.; Liang, X.; Tang, J.; and Lin, L. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *AAAI'19*.

Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5): 1160–1179.

Yu, C.; Liu, J.; Nemati, S.; and Yin, G. 2021. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1): 1–36.