Data and Text Mining

DxFormer: A Decoupled Automatic Diagnostic System Based on Decoder-Encoder Transformer with Dense Symptom Representations

Wei Chen¹, Cheng Zhong¹, Jiajie Peng²,* and Zhongyu Wei¹,²,*

¹School of Data Science, Fudan University, Shanghai, 200433, China and ¹²Research Institute of automatic and Complex Systems, Fudan University, Shanghai, 200433, China.

*To whom correspondence should be addressed.

Abstract

Motivation: Diagnosis-oriented dialogue system queries the patient’s health condition and makes predictions about possible diseases through continuous interaction with the patient. A few studies use reinforcement learning (RL) to learn the optimal policy from the joint action space of symptoms and diseases. However, existing RL (or Non-RL) methods cannot achieve sufficiently good prediction accuracy, still far from its upper limit. To address the problem, we propose a decoupled automatic diagnostic framework DxFormer, which divides the diagnosis process into two steps: symptom inquiry and disease diagnosis, where the transition from symptom inquiry to disease diagnosis is explicitly determined by the stopping criteria. In DxFormer, we treat each symptom as a token, and formalize the symptom inquiry and disease diagnosis to a language generation model and a sequence classification model respectively. We use the inverted version of Transformer, i.e., the decoder-encoder structure, to learn the representation of symptoms by jointly optimizing the reinforce reward and cross entropy loss.

Results: Extensive experiments on three public real-world datasets prove that our proposed model can effectively learn doctors’ clinical experience and achieve the state-of-the-art results in terms of symptom recall and diagnostic accuracy.

Availability: Both code and data is available from https://github.com/lemuria-wchen/DxFormer

Contact: chenwei18@fudan.edu.cn

1 Introduction

The combination of the internet and healthcare has excellent benefits and far-reaching positive effects in improving service efficiency and promoting social equity. Automated disease diagnosis is one of the rising needs in this new healthcare model, the goal of which is to simulate the actual diagnosis process of doctors.

The process of disease diagnosis can be considered as a sequence of queries and answers. Doctors choose relevant questions to ask the patient to have a better understanding of the patient’s physical condition (Martinez et al., 2018; Jian et al., 2020). In symptom-based automatic disease diagnosis, the agent has two types of actions: one type is to inquiry the patient about the possible symptoms, and another type is to predict the patient’s disease (Deng et al., 2018). The process of disease diagnosis includes several turns of symptom inquiry and a final turn of disease diagnosis.

Recent years have witnessed an emerging trend of research on the task of automatic diagnosis. Considering its interactive nature, most researchers explore to model the problem by reinforcement learning (RL) (Wei et al., 2018; Xu et al., 2019; Liao et al., 2020). At each turn of interaction, the agent chooses an action from the action space of all symptoms and diseases. Correct symptom inquiries and disease diagnoses are positively rewarded, and the policy is learned by maximizing the expected cumulative reward.

Although RL-based approaches have made progresses for developing automatic diagnostic agents, they lack of exploration on symptom modeling and result in poor symptom recall. As shown in Figure 1, the
symptom recall of most systems is within 30%. In these systems, the agent often only asks the patient for one or two symptoms and rushes to make a diagnosis. In this case, the system approximately degenerates into a pure disease classifier, and insufficient recalled symptoms also cause the diagnostic accuracy far below its upper bound.

We believe that traditional RL approaches do not perform well in symptom recall for two reasons: 1) In traditional RL framework, the action spaces of symptom inquiry and disease diagnosis are coupled (or joint), from where the agent selects an action at each step. Once the agent’s chosen action is a disease, the entire session is terminated. The agent may be “afraid” of asking wrong symptoms that get negative reward, and make a diagnosis eagerly, thus converging to a local optimal solution. Under the setting of traditional RL approaches, the number of interactive turns is determined by the reward and training settings, and it is not controllable; 2) In most past studies, symptoms are one-hot encoded as separate categories and simple MLP framework is utilized for policy learning. This simple setting is difficult to handle high-dimensional action space. Moreover, as the input features of the disease classifier, the increase of the number of symptoms will lead to a more sparse feature space, which may become intractable for the disease classifier.

To alleviate the above problems, we propose DxFormer, a decoupled automatic diagnostic framework, which consists of a Transformer-based decoder-encoder structure (not encoder-decoder). The decoder is for symptom inquiry, where the symptoms are treated as tokens or words in natural language, and the symptom inquiry is modeled as a conditional text generation task. The encoder is for disease diagnosis, where the symptoms obtained by the decoder are fed as the input sequence of the encoder, and the disease diagnosis is modeled as a sequence classification task. The decoder is encouraged to discover implicit symptoms, and the encoder is encouraged to make correct diagnosis, the two of which can work together in a decoupled manner and be trained simultaneously with little interference with each other. The dense representation of symptoms are learnt by optimizing the joint objectives. At runtime, the termination of symptom inquiry is determined by the stopping criterion. Specifically, the agent only switches from symptom inquiry to disease diagnosis if the confidence of the encoder on disease diagnosis reaches a certain threshold or the maximum number of turns is reached.

To evaluate DxFormer, we conduct experiments on three real-world structured medical dialogue datasets: Dry, MZ-4 and MZ-10. Experimental results verify that DxFormer greatly improves symptom recall and diagnostic accuracy compared with the previous state-of-the-art methods. We also conduct additional ablation experiments to demonstrate the effectiveness of the components of DxFormer and further discuss the impact of the maximum number of turns and stopping criterion threshold on the model performance.

The main contributions of this paper can be summarized as follows: 1) We propose DxFormer, a decoupled system for automatic diagnosis based on inverted Transformer, where dense representations of symptoms can be learnt by optimizing the joint objectives of decoder and encoder; 2) We discuss the impact of the maximum number of turns and stopping criterion threshold on the model performance and suggest ways to utilize DxFormer in practice; 3) Extensive experiments show that the proposed model achieves the new state-of-the-art (SOTA) results in all the three public datasets.

2 Preliminary
2.1 Formalization
In this section, we will introduce several key concepts related to automatic diagnostic system to facilitate the reader’s understanding of our motivations.

MCR In practice, annotated structured Medical Consultation Records (MCRs) are utilized to build automatic diagnostic systems [Wen et al., 2018; Zeng et al., 2020]. There are a large number of MCRs organized by disease categories available in online medical communities, such as Haodafu. Generally, each MCR consists of the patient’s self-reported report (i.e., self-report), multi-turn doctor-patient dialogue and the corresponding disease category. The self-report can be viewed as the first sentence in the dialogue.

Symptom Attribute Symptoms are widely present in actual doctor-patient conversations, they are the main topics discussed in medical dialogues and important basis for doctors to make diagnosis [Zeng et al., 2020]. However, symptoms alone in the dialogue are less informative, additional annotations are needed to find the relationship between symptoms and patients. Generally, there are two kinds of relationships between a certain symptom and the patient: 1) Positive (POS): the patient is sure to have the symptom; 2) Negative (NEG): the patient is sure not to have the symptom. The annotator is required to find out all symptom entities mentioned in the dialogue, and identify their relationship with the patient. In this paper, we refer to this relationship as the Attribute of the symptom.

Structured MCR Let $S$ denotes the set of all possible symptoms, $D$ denotes the set of possible diseases, and $A$ denotes the set of possible attributes. A structured MCR can then be denoted as: $\{(s_1, a_1), (s_2, a_2), \ldots , (s_n, a_n)\}$, where $s_i \in S$ is the i-th symptom that appears in the dialogue (with the self-report as the first utterance), $a_i \in A$ is the corresponding attribute of $s_i$, and $d \in D$ is the disease label.

Explicit & Implicit Symptoms Generally, symptoms appearing in the self-report are regarded as explicit symptoms while the others are implicit symptoms. As a notation, we take the first $k$ symptoms as explicit symptoms, denoted as $S_{exp} = \{s_1, \ldots , s_k\}$, and the implicit symptoms are denoted as $S_{imp} = \{s_{k+1}, \ldots , s_n\}$. $k$ is usually small because patients usually mention only 1 or 2 symptoms in their self-reports. Implicit symptoms are unknown during inference, thus the agent needs to find as
many implicit symptoms as possible to obtain a more complete symptom profile about the patient before making a diagnosis.

**Patient Simulator** We denote the patient simulator as $\mathcal{P}$, and $\mathcal{P}$ can be viewed as a function whose input is any symptom and whose output is the attribute of that symptom of the patient. Note that if the symptom is not among the implicit symptoms, a Unknown (UNK) attribute is returned.

**Agent** Given a patient’s explicit symptoms $S_{\text{exp}}$ and their attributes, the task of the agent is to choose a symptom from $S$ to interact with patient simulator $\mathcal{P}$, and choose the next symptom after receiving the feedback, and so on for several turns. The dialogue will terminate when the agent finally makes a diagnosis, i.e. selects a disease from $D$. The goal of the agent is to learn a policy, which can efficiently find implicit symptoms to obtain more complete information of the patient and finally make a correct diagnosis.

### 2.2 Accuracy Bound

To explain our motivation, we discuss how to evaluate automated diagnostic systems. In this section, we introduce the two components of DxFormer, namely the decoder for symptom inquiry and the encoder for disease diagnosis.

#### 3.1 Decoder for Symptom Inquiry

For symptom recall ($SX$-$Rec$), it refers to the proportion of implicit symptoms that inquired by the agent. For more clarity, assuming that for a patient, the sequence of symptoms asked by the agent is $S_{\text{agt}}$, then

$$SX-Rec = \frac{\sum |S_{\text{agt}} \cap S_{\text{imp}}|}{\sum |S_{\text{imp}}|}$$

$SX$-$Rec$ measures the agent’s ability to find implicit symptoms, and has a value between 0 and 1.

For diagnostic accuracy ($DX$-$Acc$), since making a correct diagnosis is the most important step for the agent, $DX$-$Acc$ is the final metric we aim to improve. Notably, for a disease classifier, the input is any symptom and $\mathcal{P}$ returns the attribute of that symptom of the patient. If Rec equals to 0, only explicit symptoms $S_{\text{exp}}$ can be utilized as features, we call the accuracy of the system in this case the accuracy lower bound ($Acc-LB$), when Rec is 1, both explicit symptoms and all implicit symptoms can be utilized as features, in this case, the accuracy of the system is called the accuracy upper bound ($Acc-UB$).

We present the accuracy bounds for the three datasets (Dxy, MZ-4 and MZ-10) in Table 1 as a rough reference. We train support vector machine (SVM) classifiers using five-fold cross-validation on the training set and report the accuracy on the test set. Among them, Acc-LB means that only explicit symptoms are used as features, Acc-UB means that all symptoms are used, Acc-UB ($P$) means that explicit symptoms and positive implicit symptoms are used, and Acc-UB ($N$) means that explicit symptoms and negative implicit symptoms are used.

The results in Table 1 illustrate that the diagnostic accuracy of previous SOTA systems is far from the accuracy upper bound of SVM classifier, especially for MZ-10 (Figure 1). It also suggests that both positive and negative implicit symptoms are useful for disease diagnosis. Besides, the results also give a non-rigorous reference for the agent, that is, the accuracy of a reasonable agent should be roughly between Acc-LB and Acc-UB.

The above discussion leads to the core motivation of this paper, which is to improve the diagnostic accuracy ($DX$-$Acc$) via improving the symptom recall ($SX$-$Rec$).

#### 3.2 Input Representation

Dense input representation is designed in this work. For each symptom, its input embedding is the sum of corresponding symptom, attribute, and position embeddings. One visual example is shown in Figure 2 (a). For position embeddings, we adopt sinusoidal position encoding used in [Bengio et al. 2018](#ref). The input sequence of symptoms is the concatenation of explicit symptoms and agent’s historical asked symptoms.

#### 3.3 RL formalization

As the decoder for symptom inquiry, our goal is to enhance the symptom recall within a certain number of turns, which is different from the objective function used in language models that maximizes the conditional likelihood. We can cast our generative model in the RL terminology as in [Bancino et al. 2019](#ref), due to the non-derivable property of the objective function. Our Transformer decoder can be viewed as the agent that interacts with patient simulator $\mathcal{P}$, whose parameters $\theta$ define a policy $\pi_{\theta}$, that results in an action that predict the next possible symptom based on known symptoms.

#### 3.4 Reward Setting

Considering factors such as efficiency and rationality of symptom inquiry, we design the following reward mechanism.

**Prior Reward** A particular disease is often related to a certain group of symptoms rather than all symptoms. Therefore, agent is encouraged to ask about symptoms related to the specific disease. We achieve this through the disease-symptom co-occurrence frequency matrix in training set. For each symptom in $S_{\text{exp}}$, if the frequency of co-occurrence of the symptom and the disease corresponding to the

| Dataset | $Acc-LB$ | $Acc-UB$ | $Acc-UB (P)$ | $Acc-UB (N)$ |
|---------|---------|---------|-------------|-------------|
| Dxy     | 0.644   | 0.836   | 0.808       | 0.663       |
| MZ-4    | 0.646   | 0.757   | 0.698       | 0.693       |
| MZ-10   | 0.501   | 0.706   | 0.629       | 0.618       |
case exceeds a certain threshold, a positive reward of +1 will be given, otherwise a negative reward of −1 will be given.

**Ground Reward** The agent is encouraged to inquire implicit symptoms to increase symptoms recall. For any symptom in $S_{agt}$, if the symptom is also in $S_{imp}$, a reward of +2.5 will be given, otherwise −0.5 reward will be given.

Prior Reward aims to prevent the agent from asking unrelated strange symptoms while ground reward facilitates the discovery of implicit symptoms.

### 3.1.5 Training Objective

The final reward of each action in $S_{agt}$ is equal to the sum of these two rewards, and the training objective of the decoder for symptom inquiry is formalized to minimize the negative expected reward:

$$L(\theta) = -E_{r \sim p_{theta}[R(\tau)]}$$

where $\tau = \{r_1, r_2, ..., r_m\}$ is the random reward sequence parameterized by $p_{theta}$ and $R(\tau) = \sum_{i=1}^{m} r_i$.

To compute the gradient $\nabla_{\theta} L(\theta)$, we use the REINFORCE algorithm [Williams, 1992], which approximates the reward function using a single Monte-Carlo sample from $p_{theta}$ for each training example in the minibatch. Similar to traditional text generation, we initialize the parameter $\theta$ of the decoder by language model pre-training with maximum likelihood objectives.

### 3.2 Encoder for Disease Diagnosis

We de-couple disease diagnosis from symptom inquiry for the following reasons: 1) The setting of reward is difficult in a coupled system, where the agent may struggle with whether to continue asking about symptoms or make a diagnosis. 2) There may be quite a few unknown (UNK) symptoms during the symptom inquiry, and these symptoms may interfere with the disease classifier.

#### 3.2.1 Architecture

Upon the termination of symptom inquiry, we will extracted all positive and negative symptoms obtained by the agent. We adopt a multi-layer Transformer encoder to encode these symptoms, and then pass through an average pooling layer and a linear layer to produce an output distribution over target diseases. As shown in Figure 2(b), the Transformer encoder in our disease classifier differs from the decoder for symptom inquiry in the following three ways: 1) The encoder adopts a bidirectional transformer, while the decoder is unidirectional. 2) Position embeddings are removed from the input representation in the encoder, since intuitively the disease classifier should be insensitive with the order of symptoms. 3) Our encoder is shallower than the decoder ($M < N$), the main reason is because the symptom inquiry is more complex and requires more parameters. It is worth noting that our encoder and decoder share the parameters of symptom embedding and attribute embedding.

#### 3.2.2 Training

In $D_{symptom}$, the encoder is jointly trained with the decoder. Given the explicit symptoms of a patient, we obtain $S_{sym}$ and $S_{sym}$ by the decoder using sampling decoding and greedy decoding respectively, where $S_{sym}$ is used to compute the REINFORCE reward, and $S_{sym}$ is used to compute the cross-entropy loss between the predicted disease distribution and the real disease distribution. The final loss function equals to the sum of negative REINFORCE reward and the cross-entropy loss. The parameters are initialized by pre-training the decoder and encoder using the ground truth implicit symptoms simultaneously.

#### 3.2.3 Stopping Criterion

The transition from decoder to encoder is controlled by the stopping criterion. During training, we specify a maximum turn $T_{max}$ for symptom inquiry, and the agent’s goal is to find all symptoms as much as possible within $T_{max}$ turns and make a correct diagnosis. However, asking $T_{max}$ turns each time is inefficient, because it is possible that key symptom information may have already been found, and continuing to ask the patient to obtain some unknown or insignificant symptoms is unnecessary.

We take a simple but effective method to stop the symptom early. After each turn of symptom inquiry, the currently obtained symptom information is fed to the encoder to obtain the probability distribution over the possible diseases. The symptom inquiry will be terminated once the the probability of the chosen disease is beyond the certain threshold $\epsilon$. We will analyze the effect of the threshold $\epsilon$ in 

---

**Table 2. Data statistics of Dxy, MZ-4 and MZ-10, the values in the form “a/b” in the last two columns represent the average and maximum values, respectively.**

| Dataset      | # of Samples | # of Diseases | # of Symptoms | # of Exp-Symptoms | # of Imp-Symptoms |
|--------------|--------------|---------------|---------------|-------------------|-------------------|
| Dxy (Xu et al., 2019) | 527          | 5             | 41            | 3.1 / 7           | 1.7 / 6           |
| MZ-4 (Wei et al., 2018) | 1,733       | 4             | 230           | 2.1 / 10          | 5.5 / 21          |
| MZ-10 (Chen et al., 2022) | 4,116       | 10            | 331           | 1.7 / 12          | 6.6 / 25          |

---

**Fig. 2. DiFormer is a composed of a decoder-encoder structure, where the decoder is used for symptom inquiry, and the encoder is used for disease diagnosis.**

(a) Decoder for symptom inquiry

(b) Encoder for Disease diagnosis
Table 3. Experimental results of DxFormer on Dxy, MZ-4 and MZ-10 dataset.

| Model          | Dxy         | MZ-4         | MZ-10         |
|----------------|-------------|--------------|--------------|
|                | SX-Rec | DX-Acc | # Turns | SX-Rec | DX-Acc | # Turns | SX-Rec | DX-Acc | # Turns |
| Baselines      |          |          |         |          |          |         |          |          |         |
| Flat-DQN       | 0.399    | 0.731   | 5.65    | 0.177    | 0.678   | 4.61    | 0.279    | 0.500   | 1.78    |
| REFUEL         | 0.186    | 0.721   | 3.11    | 0.215    | 0.716   | 5.01    | 0.262    | 0.505   | 5.50    |
| KR-DQN         | 0.399    | 0.740   | 5.65    | 0.177    | 0.678   | 4.61    | 0.279    | 0.485   | 5.95    |
| GAMP           | 0.268    | 0.731   | 2.84    | 0.107    | 0.644   | 2.93    | 0.067    | 0.500   | 1.78    |
| BSODA          | -        | 0.802   | -       | -        | 0.731   | -       | -        | -       | -       |
| **DxFormer (ours)** | **0.519** | **0.837** | **6.32** | **0.486** | **0.751** | **8.74** | **0.451** | **0.642** | **7.58** |

Ablation studies

| Model          | SX-Rec | DX-Acc | # Turns | SX-Rec | DX-Acc | # Turns | SX-Rec | DX-Acc | # Turns |
|----------------|--------|--------|---------|--------|--------|---------|--------|--------|---------|
| DxFormer-Sparse | 0.456  | 0.808  | 7.19    | 0.396  | 0.722  | 9.37    | 0.365  | 0.619  | 8.24    |
| DxFormer-SVM   | 0.519  | 0.789  | 6.32    | 0.486  | 0.718  | 8.74    | 0.451  | 0.603  | 7.58    |

4 Experiments

4.1 Experimental Datasets

We evaluate DxFormer on three public real-world medical dialogue datasets: Dxy, MZ-4 and MZ-10, all of which consists of a number of annotated structured MCRs described in § 2.1. Details statistics of the datasets are listed in Table 2. MZ-4 [Wen et al., 2018] is the first human-labeled dataset collected from the pediatric department of Baidu Muzhi Doctor website to evaluate automatic diagnostic system. MZ-4 includes 4 diagnosed diseases: children’s bronchitis, children’s functional dyspepsia, infantile diarrhea infection, and upper respiratory infection. Dxy [Xu et al., 2019] is an annotated medical dialog dataset collected from Dingxiang Doctor, a prevalent Chinese online healthcare website. Dxy includes 5 diagnosed diseases: allergic rhinitis, upper respiratory infection, pneumonia, children hand-foot-mouth disease, and pediatric diarrhea. MZ-10 [Chen et al., 2022] is a dataset with multi-level annotations expanded from MZ-4 to include 10 diseases, including typical diseases of digestive system, respiratory system and endocrine system. MZ-10 also contains more symptoms.

"It is worth noting that MZ-4 and MZ-10 contain an average of 5.5 and 6.6 implicit symptoms respectively, and at most more than 20 implicit symptoms, significantly more than Dxy. This suggests that the difficulty of symptom inquiry is higher on the MZ-4 and MZ-10 datasets."

4.2 Baselines

We compare DxFormer with some state-of-the-art models for automatic disease diagnosis that use different techniques, including reinforcement learning (RL), generative adversarial network (GAN), and variational autoencoder (VAE).

**DQN** [Wen et al., 2018] is an agent based on the Deep Q Network (DQN) algorithm that adopts the joint action space of symptoms and diseases, where positive reward is given to the agent at the end of a successful diagnosis.

**REFUEL** [Peng et al., 2018] is a policy based RL method with reward shaping and feature rebuilding, where a branch to reconstruct the symptom vector is utilized to guide the policy gradient.

**KR-DQN** [Xu et al., 2019] is an improved RL method based on DQN that integrates relational refinement branches and knowledge-rounded graphs to strengthen the relation between diseases and symptoms.

**GAMP** [Xia et al., 2020] is a GAN-based policy gradient network. GAMP uses the GAN network to avoid generating randomized trials of symptom, and add mutual information to encourage the model to select the most discriminative symptoms.

**BSODA** [He et al., 2022] is an non-RL bipartite framework that uses an information-theoretic reward to collect symptoms, and a multimodal variational autoencoder (MVVAE) model is used for disease prediction with a two-step sampling strategy.

"We use the open source implementation for DQN, REFUEL, KR-DQN and GAMP, since none of these papers provide any official repos or codes, and symptom recall is also not reported in most papers."

4.3 Model Configuration

DxFormer is composed of a 4-layer decoder and a 1-layer encoder. For Dxy, the embedding and hidden size is set to 128, and feed-forward size is set to 256. For MZ-4 and MZ-10, the embedding and hidden size is set to 512, and feed-forward size is set to 1024. We use Adam optimizer [Kingma and Ba, 2014] with a learning rate of $1 \times 10^{-4}$ for maximum-likelihood pre-training, and learning rate of $1 \times 10^{-4}$ for RL training. All our experiments are performed on 4 Nvidia Tesla V100-32G GPUs. Following the conventional setting, all baseline models as well as our DxFormer specify the maximum number of turns for symptom inquiry to 10, the threshold $\epsilon$ is set to 0.99.

4.4 Overall Performance

In Table 3 we compare DxFormer with the baseline models. It can be seen that under the same setting, DxFormer greatly improves the symptom recall. Compared to the baseline model with the highest recall, SX-Rec has been improved by nearly 12–27% absolute value on the three datasets. Besides, the diagnostic accuracy of DxFormer also surpasses all previous SOTAs. In particular, on MZ-10, DxFormer improves the accuracy by about 14 absolute percentages over the best baseline model.

"Considering the performance of these baseline models on the MZ-10 dataset, the diagnostic accuracy is only on par with the lower bound of the SVM classifier, which suggests that these systems approximately..."
degenerate into disease classifiers that are very weak at finding implicit symptoms. This illustrates the advantages of DxFormer over other systems when faced with more diseases and symptoms.

It is worth noting that in DxFormer, the average number of turns (\# Turns) is higher than most systems, suggesting that the model performance may not be Pareto optimal. However, as we introduced in the introduction, the average number of turns is not controllable in traditional RL-based methods. It is difficult for us to compare the model performance with the same number of turns. Nevertheless, from a clinical point of view, one would expect that asking 6-9 symptoms on average to improve diagnostic accuracy is acceptable. A slightly larger number of turns is actually an advantage since finding enough implicit symptoms must come at the expense of increasing the number of turns.

4.5 Ablation Studies

To verify the effectiveness of each component in DxFormer, we conduct some ablation experiments. **DxFormer-Sparse** is an agent that exactly the same as DxFormer except for the input presentation. DxFormer-Sparse uses the one-hot representation of symptoms and attribute and the concatenation of the one-hot vectors are fed as the input; **DxFormer-SVM** is an agent that uses the same decoder of DxFormer but replaces the encoder with a SVM classifier.

From the ablation analysis results in Table 4, both DxFormer-Sparse and DxFormer-SVM perform worse than DxFormer, which shows that the dense representation of symptoms is effective. Notably, the two variants of DxFormer can still beat SOTAs, illustrating the effectiveness of our decoupled framework. In fact, in our early attempts, models like RNN-MLP, LSTM-MLP can also work well, although not as well as DxFormer. The decoder-encoder framework and stopping criterion together contribute to the excellent performance of DxFormer.

4.6 Effect of Max Number of Turns

In figure 3(b) we analyze the effect of \( T_{\text{max}} \) on model performance on the MZ-4 and MZ-10 datasets. We find a promising result that both SX-Rec and DX-Acc increase when \( T_{\text{max}} \) increases. This confirms the core motivation of this paper, that it is feasible to improve the diagnostic accuracy by increasing the symptom recall. It can also be found that when SX-Rec reaches a certain value (about 70%), DX-Acc gradually converges to the upper bound, which is significantly higher than the upper bound of SVM classifier. On the MZ-4 and MZ-10 datasets, when the agent is allowed to interact with the patient more than 20 times, the diagnostic accuracy can be as high as about 78% and 70%, respectively, which is much higher than the current SOTA results.

We argue that the gap of the upper bound comes from two aspects: 1) the SVM classifier is based on sparse representation of symptoms rather than dense representation; 2) the SVM classifier does not exploit the sequential feature in the symptom inquiry. Besides, SX-Rec of DxFormer is also higher than that of a strong rule-based agent we create additionally for comparison. These results are encouraging and illustrate the advantage of DxFormer tremendously.

4.7 Effect of Stopping Criterion Threshold

Once the maximum number of turns \( T_{\text{max}} \) is chosen and fixed, we can further control the balance of accuracy and efficiency through the threshold \( \epsilon \) in the stopping criterion. We demonstrate in Figure 4 the effect of the threshold \( \epsilon \) on diagnostic accuracy (DX-Acc) and the average number of turns (\# Turns) on the MZ-4 and MZ-10 datasets, given \( T_{\text{max}} = 10 \). It is found that both DX-Acc and \# Turns tend to decrease as the threshold gradually decreased from 1.0 to 0.9, while the downward trend of DX-Acc is more gradual, especially in the beginning period. This suggests that it is possible to choose an appropriate threshold that reduces the average number of turns with little loss of diagnostic accuracy.

In DxFormer, \( T_{\text{max}} \) and \( \epsilon \) together determine the balance of accuracy and efficiency. In practice, we recommend to first choose the \( T_{\text{max}} \) that allows DX-Acc to converge to the upper bound, and then select an appropriate threshold to achieve acceptable accuracy and efficiency.

5 Related Work

**Automatic Disease Diagnosis** Deep reinforcement learning [Mnih et al. 2015; Silver et al. 2016] has been applied for automatic diagnosis [Tang et al. 2016; Kao et al. 2018; Peng et al. 2018] proposed reward shaping and feature rebuilding method for fast disease diagnosis. However, their data used is simulated that cannot reflect the situation of the real diagnosis. For the medical dialogue system for automatic diagnosis,
In this work, we propose a decoupled system for automatic diagnosis called DxFormer, it can learn dense representations of symptoms by simulating real medical consultation. There are also some related studies based on non-RL frameworks [Xia et al. 2020; He et al. 2022].

RL based Text Generation RL is also a popular alternative in text generation, especially when the training objective is not the traditional maximum likelihood. Ranzato et al. [2015] use REINFORCE algorithm to maximize the BLEU of the generated sequence to solve the exposure bias problem of the traditional seq2seq model. Li et al. [2016] use policy gradient algorithm to maximize the mutual information of generated response in dialogue system. Rennie et al. [2017] propose self-critical sequence training (SCST), which utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences in image captioning.

6 Conclusions and Future Work

In this work, we propose a decoupled system for automatic diagnosis called DxFormer, it can learn dense representations of symptoms by simulating real medical consultation. There are also some related studies based on non-RL frameworks [Xia et al. 2020; He et al. 2022].

7 Funding

This work is partially supported by Natural Science Foundation of China (No.21950171, No.62176205), Science and Technology Commission of Shanghai Municipality Grant (No.20D1201060, 21QA1400600) and Zhejiang Lab (No.2019KD0AD01).

Conflict of Interest: The authors declare no competing interests.

References

Bengio, Y. et al. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Febr), 1137–1155.

Chen, W. et al. (2022). A benchmark for automatic medical consultation system: Frameworks, tasks and datasets. arXiv preprint arXiv:2204.08997.

Cortes, C. and Vapnik, V. (1995). Support vector machine. Machine learning, 20(3), 273–297.

Devlin, J. et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

He, W. et al. (2022). Bsoda: A bipartite scalable framework for online disease diagnosis. In Proceedings of the ACM Web Conference 2022, pages 2511–2521.

Janisch, J. et al. (2020). Classification with costly features as a sequential decision-making problem. Machine Learning, 109(8), 1587–1615.

Kao, H.-C. et al. (2018). Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, J. et al. (2016). Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541.

Liao, K. et al. (2020). Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning. arXiv preprint arXiv:2004.14254.

Martinez, C. et al. (2020). Adaptive early classification of temporal sequences using deep reinforcement learning. Knowledge-Based Systems, 190, 105290.

Mnih, V. et al. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Peng, Y.-S. et al. (2018). Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis. Advances in neural information processing systems, 31, 7322–7331.

Radford, A. et al. (2018). Improving language understanding by generative pre-training.

Ranzato, M. et al. (2015). Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06572.

Rennie, S. J. et al. (2017). Self-critical sequence training for image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7008–7024.

Silver, D. et al. (2016). Mastering the game of go with deep neural networks and tree search. nature, 529(7587), 484–489.

Tang, K.-F. et al. (2016). Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In NIPS Workshop on Deep Reinforcement Learning.

Wei, Z. et al. (2018). Task-oriented dialogue system for automatic diagnosis. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 201–207.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3), 229–256.

Xia, Y. et al. (2020). Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 1062–1069.

Xu, L. et al. (2019). End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7346–7353.

Yu, C. et al. (2019). Reinforcement learning in healthcare: A survey. arXiv preprint arXiv:1908.06796.

Zeng, G. et al. (2020). Meddialog: A large-scale medical dialogue dataset. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9241–9250.