Bioinformatics, 2022, 1–7
https://doi.org/10.1093/bioinformatics/btac744
Advance Access Publication Date: 21 November 2022
Original Paper

Data and text mining

DxFormer: a decoupled automatic diagnostic system based on decoder–encoder transformer with dense symptom representations

Wei Chen [1], Cheng Zhong[1], Jiajie Peng[2,3]* and Zhongyu Wei[1,2]*

[1]School of Data Science, Fudan University, Shanghai 200433, China, [2]Research Institute of Automatic and Complex Systems, Fudan University, Shanghai 200433, China and [3]School of Computer Science, Northwestern Polytechnical University, Xi'an 710000, China

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on May 3, 2022; revised on October 15, 2022; editorial decision on November 10, 2022

Abstract

Motivation: Symptom-based automatic diagnostic system queries the patient's potential symptoms through continuous interaction with the patient and makes predictions about possible diseases. 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 focus on disease diagnosis while ignoring the importance of symptom inquiry. Although these systems have achieved considerable diagnostic accuracy, they are still far below its performance upper bound due to few turns of interaction with patients and insufficient performance of symptom inquiry. To address this problem, we propose a new automatic diagnostic framework called DxFormer, which decouples symptom inquiry and disease diagnosis, so that these two modules can be independently optimized. The transition from symptom inquiry to disease diagnosis is parametrically 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: We conduct experiments on three real-world medical dialogue datasets, and the experimental results verify the feasibility of increasing diagnostic accuracy by improving symptom recall. Our model overcomes the shortcomings of previous RL-based methods. By decoupling symptom query from the process of diagnosis, DxFormer greatly improves the symptom recall and achieves the state-of-the-art diagnostic accuracy.

Availability and implementation: Both code and data are available at https://github.com/lemuria-wchen/DxFormer.

Contact: jiajiepeng@nwpu.edu.cn or zywei@fudan.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

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 diagnostic process of doctors.

The process of actual 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 (Janisch *et al.*, 2020; Martinez *et al.*, 2020). In symptom-based automatic diagnostic system, the agent has two types of actions: one is to inquiry about a symptom, and another is to predict a disease (Peng *et al.*, 2018), i.e. select one of the elements from a fixed set of symptoms or diseases. The process 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; Zhong *et al.*, 2022). The RL setting of most studies is that, the agent chooses an action from the action space of all symptoms and diseases at each turn of interaction, and correct symptom inquiries and disease diagnoses are positively rewarded, then the policy can be learned by maximizing the expected cumulative reward.

Although RL-based approaches have made progresses for developing automatic diagnostic agents, most of them focus on disease diagnosis and lack of exploration on symptom modeling, resulting

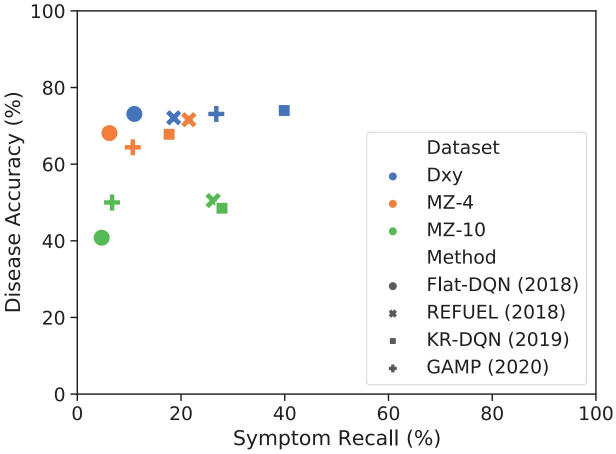


Fig. 1. Symptom Recall (%) versus Disease Accuracy (%) of previous methods including Flat-DQN, REFUEL, KR-DQN and GAMP on three datasets on three datasets Dxy, MZ-4 and MZ-10

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 practice, however, an average of 7–8 symptoms are mentioned in each conversation. This results in insufficient symptom features collected by the agent, which affects the performance of the disease diagnosis.

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 in natural language, and the symptom inquiry is modeled as a conditional text generation task. The encoder is for disease diagnosis, where the symptoms collected in symptom inquiry 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 is 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 medical dialogue datasets: Dxy, MZ-4 and MZ-10. Experimental results verify that DxFormer greatly improves symptom recall compared with the previous methods, as well as the diagnostic accuracy. 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 article can be summarized as follows: (i) we propose DxFormer, a decoupled system for automatic diagnosis based on inverted Transformer, motivated by approaching the performance upper bound of diagnostic accuracy by improving the symptom recall; (ii) 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; and (iii) extensive experiments show that the proposed model achieves the new state-of-the-art (SOTA) results in all the three real-world datasets.

2 Preliminary

2.1 Formalization

MCR: In practice, real medical consultation records (MCRs) are utilized to build data-driven automatic diagnostic system (Wei *et al.*, 2018; Zeng *et al.*, 2020). There are a large number of MCRs organized by disease categories that available in online medical communities, such as *Haodafu* (https://www.haodf.com/). Generally, each MCR consists of three parts: the patient's self-provided 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 (Lin *et al.*, 2019; 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: (i) *Positive* (POS): the patient does have the symptom; and (ii) *Negative* (NEG): the patient does not 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 article, we refer to this relationship as the *Attribute* of the symptom.

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

Explicit and 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 one or two 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_{exp} and their attributes, the task of the agent is to choose a symptom from $\mathcal{S}$ to interact with patient simulator $\mathcal{P}$, receive the feedback, choose the next symptom, and so on for several turns. The dialogue will terminate when the agent finally makes a diagnosis, i.e. selects a disease from $\mathcal{D}$. The goal of 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 earlier in this section. There are usually two metrics to evaluate the automatic diagnostic system, i.e. symptom recall and diagnostic accuracy.

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_{agt}, then

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

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

For diagnostic accuracy (DX-Acc), it refers to the proportion of correct diagnosis, and also the final metric we aim to improve. Let's further discuss the reasonable range of diagnostic accuracy. For the disease classifier, the input is the patient's symptom features finally collected by the agent, and the symptom recall determines the integrity of the symptom features. If SX-Rec equals to 0, only explicit symptoms S_{exp} can be utilized as features, we call the accuracy of the system in this case the accuracy lower bound (*Acc-LB*); if SX-Rec equals to 1, both explicit symptoms and all implicit symptoms are available as features, in this way, 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 reference. We train support vector machine (SVM) (Cortes and Vapnik, 1995) classifiers using 5-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 Acc-UB of SVM classifier, especially for MZ-10 (Fig. 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. We mention 'non-rigorous' because the performance of diagnostic accuracy of different classifiers is different. The above discussion leads to the core motivation of this article, which is to *improve the diagnostic accuracy via improving the symptom recall*.

3 Methods

3.1 Decoder for symptom inquiry

For symptom inquiry, assuming that there is no limit on the number of turns, the agent can find all implicit symptoms by simply traversing the symptoms in $\mathcal{S}$. In this case, the recall equals to 1, and the accuracy can reach the upper bound. However, the size of $\mathcal{S}$, i.e. the action space of symptom inquiry, can be potentially large, this policy is undoubtedly inefficient. Fortunately, due to the apparent cooccurrence between symptoms (Zhong *et al.*, 2022), training a more efficient agent is promising.

3.1.1 Architecture

In DxFormer, we analogize the process of symptom inquiry to a language model (Bengio *et al.*, 2003). The symptoms are regarded as tokens, and attributes of symptoms are regarded as features of tokens, then symptom inquiry can be regarded as a text generation problem. We adopt the multi-layer Transformer (Radford *et al.*, 2018) decoder as the recurrent model, which applies a multi-headed self-attention operation over the historical symptom-attribute sequence followed by position-wise feedforward layers to produce an output distribution over target symptoms.

3.1.2 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. For symptom and attribute, we utilize embedding layers to map any symptom in $\mathcal{S}$ and any attribute in $\mathcal{A}$ into dense vectors with same dimension. For position embeddings, we adopt sinusoidal position encoding used in BERT (Devlin *et al.*, 2019). The input sequence of symptoms is the concatenation of explicit symptoms and agent's historical asked symptoms.

Table 1. The accuracy bound of SVM classifier trained on Dxy, MZ-4 and MZ-10, these classifiers differ in the features they use

Dataset	Acc-LB	Acc-UB	Acc-UB (P)	Acc-UB (N)
Dxy	0.644	0.856	0.808	0.663
MZ-4	0.646	0.757	0.698	0.693
MZ-10	0.501	0.706	0.629	0.618

3.1.3 RL formalization

As the decoder for symptom inquiry, our goal is to find as many implicit symptoms as possible within a given number of interactions, which is a discrete objective different from the one used in language models that maximizes the conditional likelihood. We can cast our generative model in the RL terminology as in Ranzato *et al.* (2015) 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 θ define a policy p_θ, that results in an action that predict the next possible symptom based on known symptoms.

3.1.4 Reward setting

Priori reward: A particular disease is often related to a certain group of symptoms rather than all symptoms (Zhong *et al.*, 2022). 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_{agt}, if the frequency of co-occurrence of the symptom and the disease corresponding to the case exceeds zero, a positive reward of + 1 will be given, otherwise a negative reward of –1 will be given.

Ground reward: The agent is encouraged to inquiry 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.

Priori reward aims to prevent the agent from asking unrelated strange symptoms, while ground reward facilitates the discovery of implicit symptoms. We will discuss the settings of these reward parameters and their impact on the results in the Supplementary Material.

3.1.5 Training objective

The final reward of each action in S_{agt} is equal to the sum of priori reward and ground reward, and the training objective of the decoder is to minimize the negative expected reward:

$$\mathcal{L}(\theta) = -\mathbf{E}_{\tau\sim p_\theta}[R(\tau)] \tag{1}$$

where $\tau = \{r_1, r_2, \ldots, r_m\}$ is random reward sequence that obeys the conditional probability distribution parameterized by p_θ, and $R(\tau) = \sum_{i=1}^{m} r_i$.

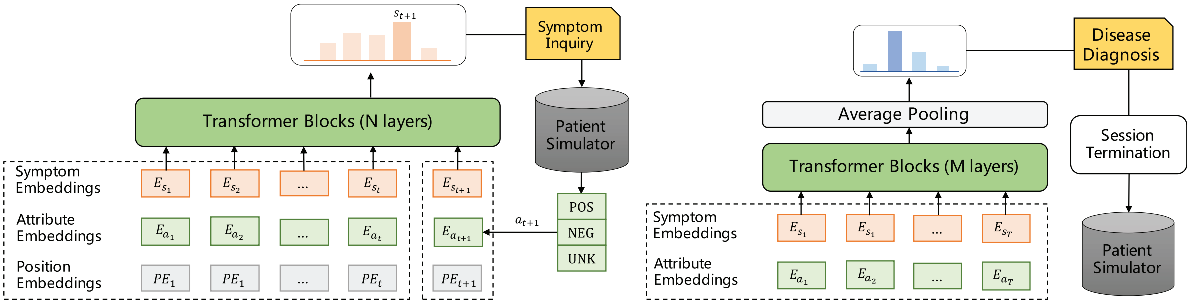
To compute the gradient $\nabla_\theta L(\theta)$, we use the REINFORCE algorithm (Williams, 1992), which approximates the reward function using a single Monte-Carlo sample from p_θ for each training example in the minibatch. Similar to traditional text generation, we initialize the parameter θ of the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

3.2 Encoder for disease diagnosis

Upon the termination of symptom inquiry, we will extracted all positive and negative symptoms obtained by the agent. These symptoms, together with explicit symptoms, are used as features for disease diagnosis.

3.2.1 Architecture

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, the Transformer encoder in our disease classifier differs from the decoder for symptom inquiry in the following three ways: (i) the encoder adopts a bidirectional transformer, while the decoder is unidirectional; (ii) position embeddings is removed from the input representation in the encoder, since intuitively the disease classifier should be insensitive with the order of symptoms; and (iii) our encoder is shallower than the decoder ($M < N$), the main reason is because the symptom inquiry is more complex and requires more



(a) Decoder for symptom inquiry (b) Encoder for Disease diagnosis

Fig. 2. DxFormer is composed of a decoder–encoder structure, where the decoder (a) is used for symptom inquiry, and the encoder (b) is used for disease diagnosis

parameters. It is worth noting that our encoder and decoder *share* the parameters of symptom embedding and attribute embedding.

3.2.2 Training

In DxFormer, the encoder is jointly trained with the decoder. Given the explicit symptoms of a patient, we obtain S_{sdec} and S_{gdec} by the decoder using sampling decoding and greedy decoding respectively, where S_{sdec} is used to compute the REINFORCE reward, and S_{gdec} 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.

3.2.3 Stopping criterion

The transition from symptom inquiry to disease diagnosis is determined by the stopping criterion. During training, we specify a maximum turn T_{max} for symptom inquiry, and the agent's goal is to find as many implicit symptoms as possible within T_{max} turns and make correct diagnosis. However, agents do not always need to inquiry symptoms T_{max} times. For some cases, the key symptom information has been collected, so it is unnecessary to continue to ask the patient to obtain some unknown or unimportant symptoms.

We take a simple but effective method to stop the symptom inquiry early. After each turn of symptom inquiry, the currently collected symptoms are fed to the encoder to compute the probability distribution over the possible diseases. The symptom inquiry will be terminated once the probability of the predicted disease (with maximum probability) is beyond the certain threshold ϵ.

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 Section 2.1. Details statistics of the datasets are listed in Table 2.

MZ-4 (Wei *et al.*, 2018): The first human-labeled dataset collected from the pediatric department of Baidu Muzhi Doctor (https://muzhi.baidu.com/) to evaluate automatic diagnostic system. MZ-4 includes four diagnosed diseases: children's bronchitis, children's functional dyspepsia, infantile diarrhea infection and upper respiratory infection.

Dxy (Xu *et al.*, 2019): An annotated medical dialog dataset collected from Dingxiang Doctor (https://dxy.com/), a prevalent Chinese online healthcare website. Dxy includes five diagnosed diseases: allergic rhinitis, upper respiratory infection, pneumonia, children hand-foot-mouth disease and pediatric diarrhea.

MZ-10: 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. The MZ-10 are annotated by medical students. Each dialogue is annotated twice, and the kappa coefficient of symptom labels is 92.71%, which represents a high consistency between the two annotations.

4.2 Baselines

We compare DxFormer with some SOTA models for automatic diagnosis that use different techniques, including RL, generative adversarial network and variational autoencoder.

DQN (Wei *et al.*, 2018): 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 success diagnosis.

REFUEL (Peng *et al.*, 2018): 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): An improved RL method based on DQN that integrates relational refinement branches and knowledge-routed graphs to strengthen the relation between diseases and symptoms.

GAMP (Xia *et al.*, 2020): 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): An non-RL bipartite framework that uses an information-theoretic reward to collect symptoms, and a multimodal variational autoencoder model is used for disease prediction with a two-step sampling strategy.

We use the open source implementation (https://github.com/Guardianzc/DISCOpen-MedBox-DialoDiagnosis) for DQN, REFUEL, KR-DQN and GAMP, since none of these papers provide official 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 3×10^{-4} for maximum-likelihood pre-training, and learning rate of 1×10^{-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 during inference, the threshold ϵ is set to 0.99. It is worth noting that during training, the maximum number of turns of DxFormer is set to 40.

4.4 Main findings

In Table 3, we report the performance of DxFormer as well as the baseline models. DxFormer shows impressive performance. On the three real-world datasets, DxFormer can find about 45–52% of implicit symptoms in 6–8 turns, and reach about 64–84% of diagnostic accuracy. Compared to best baseline models, DxFormer greatly improves the symptom recall, with an absolute improvement of

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)	1733	4	230	2.1/10	5.5/21
MZ-10	4116	10	331	1.7/12	6.6/25

Note: Exp is short for Explicit and Imp is short for Implicit.

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 (Wei *et al.*, 2018)	0.110	0.731	1.96	0.062	0.681	1.27	0.047	0.408	9.75
REFUEL (Peng *et al.*, 2018)	0.186	0.721	3.11	0.215	0.716	5.01	0.262	0.505	5.50
KR-DQN (Xu *et al.*, 2019)	0.399	0.740	5.65	0.177	0.678	4.61	0.279	0.485	5.95
GAMP (Xia *et al.*, 2020)	0.268	0.731	2.84	0.107	0.644	2.93	0.067	0.500	1.78
BSODA (He *et al.*, 2022)	–	0.802	–	–	0.731	–	–	–	–
DxFormer (ours)	**0.506**	**0.817**	6.32	**0.479**	**0.743**	8.74	**0.449**	**0.633**	7.58
Ablation studies									
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

The boldface values in the table are significantly better than the best baseline values at the 5% significance level (we repeated the experiments for 10 times and conduct one-sample t-test to compute the significance).

nearly 12–27%. Besides, the diagnostic accuracy also surpasses all previous models. In particular, on MZ-10, DxFormer improves the accuracy by about 14 absolute percentages over the best baseline.

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 (see Table 1), 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, agent interacts with patient simulator more rounds (# Turns) than most systems. It is not surprising since enough interaction turns are the guarantee of high symptom recall. As we mentioned in the introduction, the number of interactions is not controllable in traditional RL-based methods. It is difficult for us to compare the model performance within 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.

4.5 Effect of max number of turns

In Figure 3, we visualize the effect of max number of turns, i.e. T_{max}, on model performance on the MZ-4 and MZ-10 datasets. It can be seen that both symptom recall and diagnostic accuracy increase with the increase of the maximum number of allowed interactions. This confirms the core motivation of this article, that *it is feasible to improve the diagnostic accuracy by increasing the symptom recall*. It can also be found that when symptom recall exceeds about 70%, diagnostic accuracy gradually converges to the upper bound, which is 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.

Besides, we also create a strong rule-based agent for symptom inquiry, i.e. based on the symptom co-occurrence probability matrix, to find the symptom with the highest correlation with the collected symptom set each time. The yellow line in Figure 3 shows the results of the method. It can be seen that rule-based agent performs worse than DxFormer, especially on the MZ-10 dataset. When the number of symptoms increases (from MZ-4 to MZ-10), the dense representation of symptoms shows advantages, as the symptom recall of DxFormer does not decrease.

4.6 Effect of stopping criterion threshold

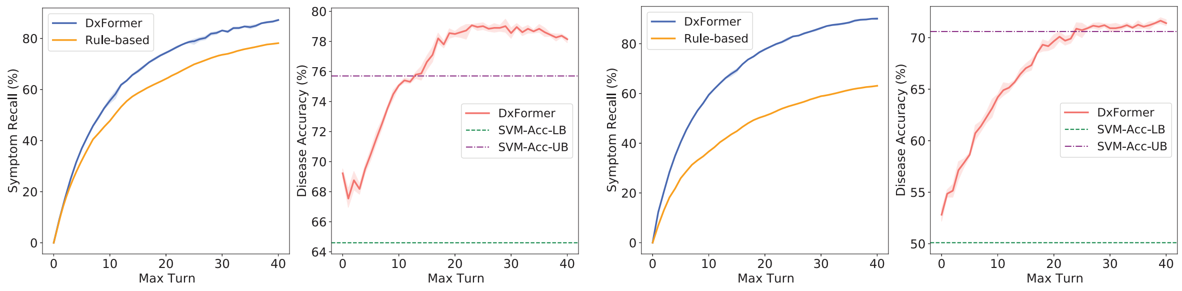
We further explore the impact of the threshold value in stopping criterion on model performance. We fix the maximum number of turns to 10 and set different thresholds to observe the model performance. The results are visualized in Figure 4. We have two findings. First, both diagnostic accuracy and average turns decrease as the threshold value decreases from 1.0 to 0.9, this shows that, as the number of turns for symptom inquiry increases, the collected symptoms increase, and the certainty of disease classifier also increases. Secondly, symptom recall declines slowly at the beginning. 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_{max} and ϵ together determine the balance of accuracy and efficiency. In practice, we recommend to first choose the T_{max} that allows diagnostic accuracy to converge to the upper bound, and then select an appropriate threshold to achieve acceptable accuracy and efficiency.

4.7 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 symptom 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 an SVM classifier.

From the ablation analysis results in Table 3, 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



(a) MZ-4 dataset

(b) MZ-10 dataset

Fig. 3. The effect of max turns on the Symptom Recall and Diagnostic Accuracy on MZ-4 (a) and MZ-10 (b) dataset

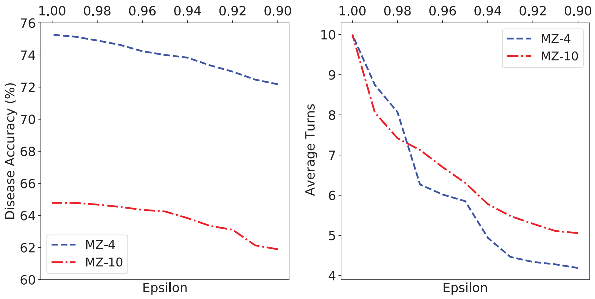


Fig. 4. The effect of threshold on MZ-4 and MZ-10 dataset

well, although not as well as DxFormer. The decoder–encoder framework and stopping criterion together contributes to the excellent performance of DxFormer.

5 Related work

Automatic disease diagnosis: Machine learning-based method plays an important role in solving medical prediction, such as depression prediction (Aekwarangkoon and Thanathamathee, 2022) and epidemic prediction (Kim, 2021). Automatic disease diagnosis is an important part of medical prediction. Deep RL (Mnih *et al.*, 2013; Silver *et al.*, 2016) has been applied for automatic diagnosis (Kao *et al.*, 2018; Tang *et al.*, 2016). 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, Wei *et al.* (2018) annotated the first medical dataset for dialogue system and use a Deep Q-network (DQN) to collect additional symptoms via conversation with patients. Xu *et al.* (2019) released another medical dataset for the dialogue system and introduce prior knowledge to improve the diagnosis accuracy. Zhong *et al.* (2022) propose a hierarchical RL framework based on master–worker structure for simulating real medical consultation. There are also some related studies based on non-RL framework (He *et al.*, 2022; Xia *et al.*, 2020). Diaformer (Chen *et al.*, 2022a) is a contemporary work with our DxFormer, which is similar to our ideas, but with different motivations. We do not compare the results because the number of turns is set differently.

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, which utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences in image captioning.

6 Conclusions, limitations and future work

In this work, we propose a decoupled system for automatic diagnosis called DxFormer, which uses Transformer's decoder and encoder for symptom query and disease diagnosis with dense symptom representations. Symptom query and disease diagnosis are formalized into conditional text generation and sequence classification. The stopping criterion controls the transition from symptom query to disease diagnosis to parametrically control the number of interaction turns. Extensive experiments show that DxFormer can greatly improve the symptom recall rate and diagnostic accuracy.

Although promising results have been obtained, the system still has some limitations: (i) in this paper, the stopping criterion is simple and crude, and its reliability and scalability should be discussed; (ii) in practice, agents should be allowed to ask multiple symptoms at one time rather than a single symptom to improve efficiency; and (iii) it is not enough to use only the symptoms for diagnosis. In the actual situation, more factors need to be considered, including medical examination, past medical history, surrounding environment, etc.

In future work, we hope to further explore automatic disease diagnosis based on more features. Besides, we will also try to improve the model from the perspective of biological interpretability.

Funding

This work was partially supported by the Natural Science Foundation of China [71991471 and 6217020551], Science and Technology Commission of Shanghai Municipality [20dz1200600 and 21QA1400600] and Zhejiang Lab [2019KD0AD01].

Conflict of Interest: none declared.

References

Aekwarangkoon,S. and Thanathamathee,P. (2022) Associated patterns and predicting model of life trauma, depression, and suicide using ensemble machine learning. *Emerg. Sci. J.*, **6**, 679–693.

Bengio,Y. *et al.* (2003) A neural probabilistic language model. *J. Mach. Learn. Res.*, **3**, 1137–1155.

Chen,J. *et al.* (2022a) Diaformer: Automatic diagnosis via symptoms sequence generation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. **36**, pp. 4432–4440.

Cortes,C. and Vapnik,V. (1995) Support vector machine. *Mach. Learn.*, **20**, 273–297.

Devlin,J. *et al.* (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 4171–4186.

He,W. *et al.* (2022) BSODA: a bipartite scalable framework for online disease diagnosis. In: *Proceedings of the ACM Web Conference 2022, Lyon, France, April 25 - 29, 2022*, pp. 2511–2521.

Janisch,J. *et al.* (2020) Classification with costly features as a sequential decision-making problem. *Mach. Learn.*, **109**, 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, Brussels, Belgium, October 31-November 4, 2018*, Vol. **32**, pp. 2305–2313.

Kim,M. (2021) Prediction of covid-19 confirmed cases after vaccination: based on statistical and deep learning models. *Sci. Med. J.*, **3**, 153–165.

Kingma,D.P. and Ba,J. (2015) Adam: A method for stochastic optimization. In: *International Conference on Learning Representations (ICLR)*.

Li,J. *et al.* (2016) Deep reinforcement learning for dialogue generation. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas*, Association for Computational Linguistics. pp. 1192–1202.

Lin,X. *et al.* (2019) Enhancing dialogue symptom diagnosis with global attention and symptom graph. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, November 5–7, 2019*, pp. 5033–5042.

Martinez,C. *et al.* (2020) Adaptive early classification of temporal sequences using deep reinforcement learning. *Knowl. Based Syst.*, **190**, 105290.

Mnih,V. *et al.* (2015) Human-level control through deep reinforcement learning. *Nature*, **518**(7540), 529–533.

Peng,Y.-S. *et al.* (2018) Refuel: exploring sparse features in deep reinforcement learning for fast disease diagnosis. *Adv. Neural Inf. Process. Syst.*, **31**, 7322–7331.

Radford,A. *et al.* (2018) *Improving Language Understanding by Generative Pre-Training*.

Ranzato,M. *et al.* (2016) Sequence level training with recurrent neural networks. Publisher Copyright: © ICLR 2016: San Juan, Puerto Rico. All Rights Reserved.; *4th International Conference on Learning Representations, ICLR 2016*.

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, CVPR 2017, Honolulu, Hawaii, July 22-July 25, 2017*, pp. 7008–7024.

Silver,D. *et al.* (2016) Mastering the game of go with deep neural networks and tree search. *Nature*, **529**, 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, Barcelona, Spain, Dec 5-10, 2016*.

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), ACL 2018, Melbourne, Australia, July 15-20, 2018*, pp. 201–207.

Williams,R.J. (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, **8**, 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, Hilton New York Midtown, New York, USA, February 7-12, 2020*, Vol. **34**, pp. 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, Hilton Hawaiian Village, Honolulu, Hawaii, USA, January 27 – February 1, 2019*, Vol. **33**, pp. 7346–7353.

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)*, pp. 9241–9250.

Zhong,C. *et al.* (2022) Hierarchical reinforcement learning for automatic disease diagnosis. *Bioinformatics*, **38**, 3995–4001.