# Intelligent Emergency Medical QA System Based on Deep Reinforcement Learning

Zihao Wang and Xuedong Chen(✉)

School of Economics and Management, Beijing Jiaotong University, Beijing, China
xdchen@bjtu.edu.cn

**Abstract.** This paper mainly focuses on solving the problem of insufficient intelligence of the current emergency medical question answering system, and proposes a solution of deep integration of question answering system and deep reinforcement learning model according to the relevant technology of natural language processing. This paper focuses on the construction and implementation of the interactive environment of deep reinforcement learning, which uses multiple pre trained language models in series, evaluates the environment through the core scoring network of the agent, and decides to return the relevant reply to the user. The structure of several pre training language models is discussed, and the conclusion that dynamic word embedding model with attention mechanism should be used as much as possible, and the complexity of output layer model should be increased.

**Keywords:** Intelligent Q&A System · Reinforcement learning · Natural language processing

## 1 Introduction

At present, intelligent question answering system plays an important role in the field of emergency medicine. However, the current market of intelligent question answering system is often through the "pseudo intelligent" way to interact with users. This paper is committed to the application of natural language processing and deep reinforcement learning technology to improve the intelligence of emergency medical Q&A system, so as to solve the problem of pseudo intelligence in the current Q&A system.

## 2 Model Architecture

### 2.1 Background

Considering that the whole process of deep reinforcement learning should be deeply integrated with the emergency medical intelligent question answering system, the working principle of the question answering system should be considered when designing the model and technical process. For example, when users use the question answering system, they input text segments of some medical related query questions, and the output


X. Shi et al. (Eds.): *LISS 2021*, LNOR, pp. 124–131, 2022.
https://doi.org/10.1007/978-981-16-8656-6_11

phase is the text data corresponding to the questions answered by users that the system should return. How to improve the quality and accuracy of the answer text returned to users is the problem that needs to be solved by applying deep reinforcement learning technology. The user's evaluation of the system return results (usually divided into several levels, such as 1–5 points) can often be used as the basis for modifying the model. In the field of deep learning, this kind of feedback can be called the supervision of the model.

## 2.2 Environment Construction

Based on the comprehensive consideration of the key points listed above, on the basis of ensuring the integrity of the Q & A system process, this paper integrates the idea of deep reinforcement learning training and application, so as to realize the effect of continuous self-training through the interaction between users and the system after the system is launched and deployed, so as to achieve the purpose of intelligent Q&A system. The overall technical implementation process is shown in Fig. 1.

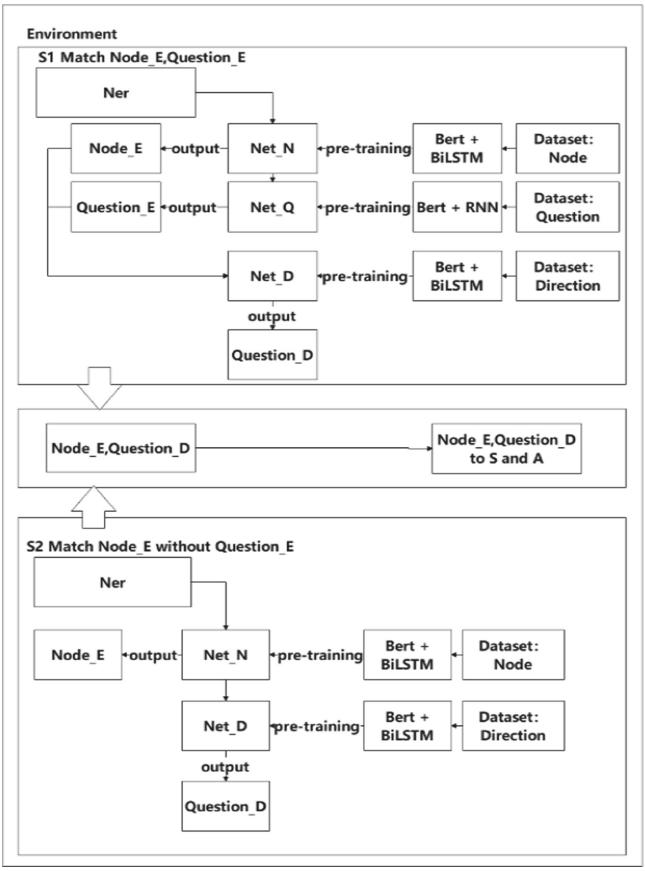


**Fig. 1.** Interactive environment structure

According to the standard implementation of reinforcement learning [7, 8], the whole system can be divided into two parts: environment part and agent part. The structure and implementation process of the agent part are relatively fixed, which will be described in Sect. 2.C. In the whole process of combining deep reinforcement learning with practical projects, it is often difficult to build an environment for interaction with agents. It is necessary to consider the interaction with business systems while considering several parameters that agents need to obtain.

Generally speaking, we need to get the four parameters *S, A, R, S′* in the environment to interact with the agent [7], so as to train the agent. For this system, *S* represents the current state of the environment and can be abstracted as the user's questions; *A* represents the next action in s state, which can abstract the solution provided by the system to the user; *R* stands for the benefit obtained by executing action a in s state, which can be abstracted as the user's evaluation of answering *A* after *S* according to the user's question; *S′* represents the next state after action *A* is executed in S state, which can be abstracted as the state after the whole process of Q&A.

In terms of process, the original question entered by the user when using the question answering system is regarded as the original initial input of the whole system. This input is a text type of data. If you want to get the corresponding parameter s which can be used to interact with the agent from this text data, you need to first do word embedding [1, 2] processing on the text, and make the problem text vectorized to facilitate the subsequent network processing.

Generally, for the original problem processing, two key information will be extracted first, namely, the medical entity *N* in the text and the query key word *Q* (ask which aspect of entity *N* is concerned). Before the original problem is processed, several corresponding language models need to be trained in advance. Firstly, the medical entity recognition model Net-N is used to identify the medical entity *N* existing in the problem. Then, the key query word recognition model Net-Q is used to identify the key query words *Q* in the original problem. There are two kinds of unnecessary situations here. One is to recognize entity *N* in the original problem, and also to identify key query word *Q*. the related expression of key query word *Q* can be transferred to the query direction model Net-D1 to obtain the query direction *D1*. Finally, the entity *N* and query direction *D1* can be abstracted into the current state *S* and transmitted to the agent. The second is that entity *N* can be identified in the original problem, but the key query word *Q* cannot be identified. Therefore, it is necessary to transfer the original problem to the query direction model Net-D2 directly after word-embedding, and then the query direction *D2* can be abstracted into the current state s and transmitted to the agent. The training strategies of these models will be described in Sect. 3.

### 2.3 Agent Structure

The agent part mainly consists of three structures, namely, Q-net scoring network, Q-net scoring network and Q-net scoring network ε-Policy greedy policy and buffer state buffer. The core goal of the agent part is to train a scoring network which can evaluate the current state and choose the next optimal strategy.

As described in Sect. 2.B, the state *S* and the query direction *D* obtained from the environment will interact with the agent. Through the scoring network in the agent,

the mapping between the state S and the query direction $D$ can be obtained. $Q$ ($S$, $D$) represents the state $S$, and then according to the $\varepsilon$ Strategy: according to the probability, choose between action $A$ and random a with the largest $Q$, and execute the score of action $A$ when the query direction is $D$. $A$ can be abstracted as the content vector returned to the user in the case of state $S$ and query direction $D$, and the content vector will be decoded before returning to the user, According to the relational query, the corresponding triple data is obtained from the previously established neo4j diagram database related to emergency medicine, and then the triple data is processed, put into the corresponding response text template and returned to the user, and the status at this time is recorded as $S'$ (Fig. 2).

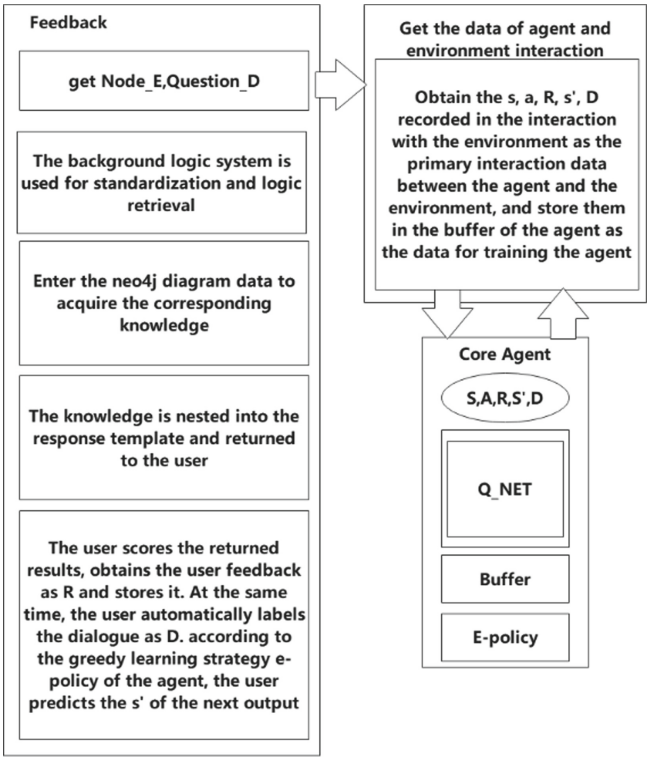


**Fig. 2.** Interaction between agent and environment.

After the user obtains the results returned by the system, he scores the returned results according to whether the results meet his own needs. The scoring results will be recorded by the agent as $R$ in the demand. Thus, all the parameters $S$, $A$, $R$, $S'$ needed for training the scoring network can be obtained, a group of parameters obtained from this round of operation are recorded as an interaction record and stored in the buffer of the agent for subsequent training. The operation basis for this is the experience playback strategy (inserting Literature) in reinforcement learning, which is used to make full use of valuable interaction data.

The process of training q-scoring network is mainly based on back propagation to update the parameters in the network. The core formula is as follows.

$$Q(St, At, w) += \alpha[Rt + 1 + \gamma maxaq^{(st+1,at,w)} - Q^{(St,At,w)}] \quad (1)$$

$$\triangle_w = \alpha(Rt + 1 + \gamma maxaq^{(st+1,at,w)} - q^{(st,st,w)}) \triangle wq^{(st,at,w)} \quad (2)$$

The whole process of interaction and parameter updating can be represented by pseudo code are as Fig. 3.

Initialize memory n in memory *D*

Initialize random weights $\theta$ Function *Q* of action value

Initialization weight $\theta$-=$\theta$ Function ^ *Q* of target action value

Initialization of pre training language model

Initialize the dictionary of node name and query keywords

For text in Query text entered by the user:

If match to node name and query word (*S1*):

Initialize the first scene *S1* processing function $\Phi$

Get *N* and *D* through Net-N,Net-D,Net-Q

Elif match to node name without query word (*S2*):

Initialize the first scene *S2* processing function $\Phi$

Get *N* and *D* through Net-N,Net-D

Else no node matched:

Break user re input

For record of agent environment interaction:

According to P($\varepsilon$) Choose a random action

Or choose a maximum value with function *Q*

Execute action and get a reward Rt from user

Let *st*+1=*st*, *at*, *XT*+1and process $\Phi$t+1=$\Phi$( *s*t+1)

Take ($\Phi$t,at,rt, $\Phi$t+1) store in D

Sample a random small batch of training in D

Perform *a* (*Q* reality-q estimate) square gradient regression to update the weights θ

Execute *a* ^ *q* = *q* for every number of steps

**Fig. 3.** Pseudo code of interaction process

The whole interaction process and business process are deeply integrated, and can be deployed on related services in reality, so that they can train themselves online. Each time the user uses the service, the background algorithm will interact with the model and the environment. The user will score the model results. This process will be stored in the memory of the agent as a historical record to provide data for the training of the model. With the increasing use of the service, the agent continuously carries out reinforcement learning training, and the accuracy is gradually improved.

## 3 Pre Training Model

In the whole process, we need to use four language models built by deep learning. Net-N, Net-Q, Net-D1 and Net-D2 need to be trained before the whole process. Considering that the difficulty of data acquisition in the early stage of the whole project leads to a small amount of data, we consider using transfer learning strategy instead of training a complete network from the beginning to the end.

Bert [6] is a pre training language model based on attention mechanism [4], which is more suitable for this project than word2vec static model (Fig. 4).

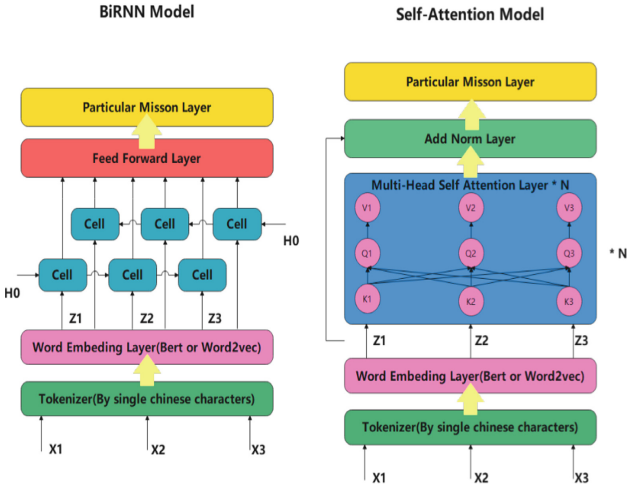


**Fig. 4.** Network structure of pre training model

Using the technology of transfer learning, the data of text type can be put into the Bert pre training model for word embedding operation. In fact, the model behind Bert can be adjusted according to the actual needs of the project. At the same time, in the process of back propagation, the parameters of the previous Bert model are not updated, and only the parameters of the later connected model are optimized.

In this paper, three models word2vec RNN, Bert birnn and Bert bilstm are tested. The confusion matrix is shown in Fig. 5, and the evaluation of the three models is shown in Table 1.

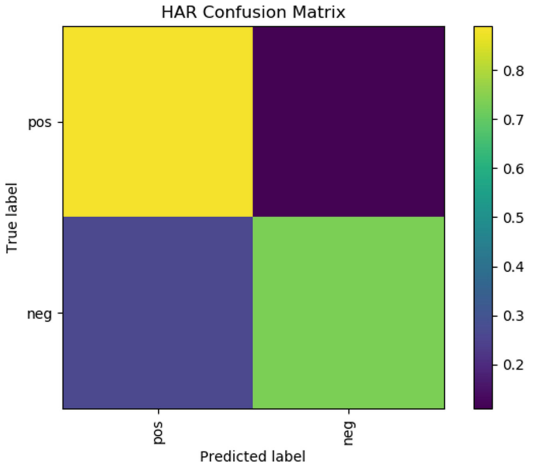


**Fig. 5.** Interaction between agent and environment

**Table 1.** Model performance comparation

| Table head | Model performance | | | |
|---|---|---|---|---|
| | Acc | Precision | Recall | F1 |
| Word2vec-BiRnn | 72.153 | 74.211 | 71.957 | 73.066 |
| Bert-Attention | 80.327 | 81.956 | 80.327 | 81.133 |
| Bert-BiRnn | 83.425 | 82.632 | 83.081 | 82.855 |

Considering that the input order of training data is random, it will affect the performance of the model, so the results of model evaluation are the average results after multiple training. It can be seen from the performance of the model tested on the test set that in the word embedding stage, the result of the Bert model based on the attention mechanism in the specific task is significantly better than that of the static model word2vec, while the performance of the output layer model adjusted according to the task requirements is positively correlated with the complexity of the model without over fitting, Later, we can consider the method of model complexity, such as adding several fully connected layers before the output layer, or increasing the number of layers in the loop network (all the models mentioned in this paper are single-layer) to improve the overall performance of the model.

## 4 Conclusion

This paper mainly proposes a set of deep intelligent solution of emergency medical intelligent Q & A, focusing on the deep integration of artificial intelligence and real scene. In the field of artificial intelligence, the overall structure is a reinforcement learning model of interaction between environment and agent, and the relevant mechanism of

the model is highly matched with the link of application business. The environment is completed by multiple language models, and the performance of several language models determines the stability of the whole environment. After many experiments, it is found that we should choose the word embedding model based on attention mechanism, and increase the complexity of the model in the output layer stage to improve the overall performance of the model.

When building the environment, this paper uses several language models in series to form the whole environment and interact with the agent. The disadvantage of this method is that with the increase of the number of language models in series, the performance of the model will gradually decline. Therefore, we can consider merging the whole environment into a whole language model in the future, so as to realize the end to end training, The attenuation of model performance is minimized.

**Acknowledgment.** This work was partly supported by the National Key Research and Development Plan for Science and Technology Winter Olympics of the Ministry of Science and Technology of China (2019YFF0302301).

## References

1. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed Representations of Words and Phrases and their Compositionality, October 2013. arXiv:1310.4546v1 [cs.CL]. https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
2. Mikolov, T., Sutskever, I., Chen, K., et al.: Efficient Estimation of Word Representations in Vector Space. Comput. Sci., October 2013. https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
3. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks, October 2014. arXiv preprint arXiv:1409.3215, https://arxiv.org/abs/1409.3215.
4. Vaswani, A., Shazeer, N.: Attention is all you need, June 2017. arXiv preprint arXiv:1706.03762,2017, https://arxiv.org/abs/1706.03762
5. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450, December 2016
6. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding, April 2018, arXiv preprint arXiv:1810.04805, https://arxiv.org/pdf/1810.04805.pdf
7. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2016, vol. 30(1), May 2017. https://arxiv.org/pdf/1606.01541.pdf
8. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization, May 2017. arXiv preprint arXiv:1705.04304, 2017, https://arxiv.org/abs/1705.04304