22 June 2024

Medical Diagnosis in Dialogue systems using Reinforcement Learning

Introduction

Medical diagnosis is the process of identifying diseases and conditions based on symptoms, history, exams, and tests. It's crucial because it guides treatment, helps prevent complications, and improves patient outcomes.

I have fever and cough

Do you suffer from headache ?

Unfortunately, yes.

You have pneumonia

Motivation

Early detection of disease using AI

- AI models can detect pancreatic cancer a median of 438 days before clinical diagnosis. (Med Xpress) (MIT News).

Lower the costs

- AI tools for diabetes management, such as continuous glucose monitors, educed the need for in-person appointments, also significantly lowered hospital readmission rates and overall healthcare expenditures. (Echelon Health).

Fast way to give you an idea on what clinic type to visit

- provide personalized recommendations on the type of clinic or specialist to visit based on the patient's specific symptoms(McKinsey & Company)

Different Ways of Medical Diagnosis

1. Rule-Based Systems

They don't improve over time without explicit human intervention and updating of rules.

2. Large Language Models

a. Generate output based on vast pre-trained *language* datasets

b. Training and deploying LLMs require significant computational resources, making them costly and less accessible

3. Reinforcement Learning

Learns from the environment based on a reward/penality

Actions

The most probable action.
In our case, it is either asking a patient about specific symptoms or diagnosing a disease.

States

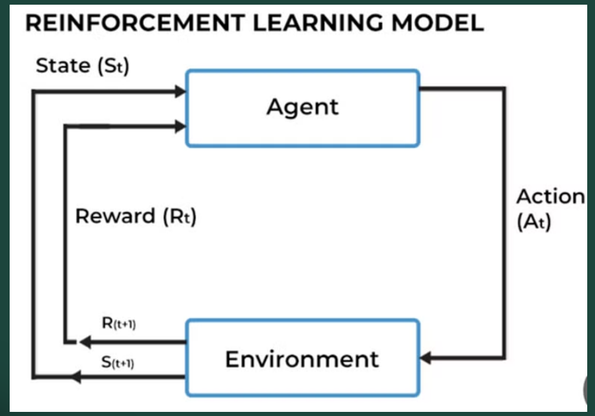
a state is the current set of symptoms that is known to agent.

Reward

are feedback given to the agent based on the effectiveness of its actions

Policy

the strategy or set of rules that the agent follows to decide its actions in each state



Proximal Policy Optimization

a. It is the state-of-art algorithm for reinforcement learning

b. Considered on-policy algorithm

c. Train the policy directly

Actor

Critic

Proximal Policy Optimization

a. It is the state-of-art algorithm for reinforcement learning

b. Considered on-policy algorithm

c. Train the policy directly

There are multiple limitations for PPO. However, the most prominent limitation is Hyperparameter Sensitivity.

Actor

Critic

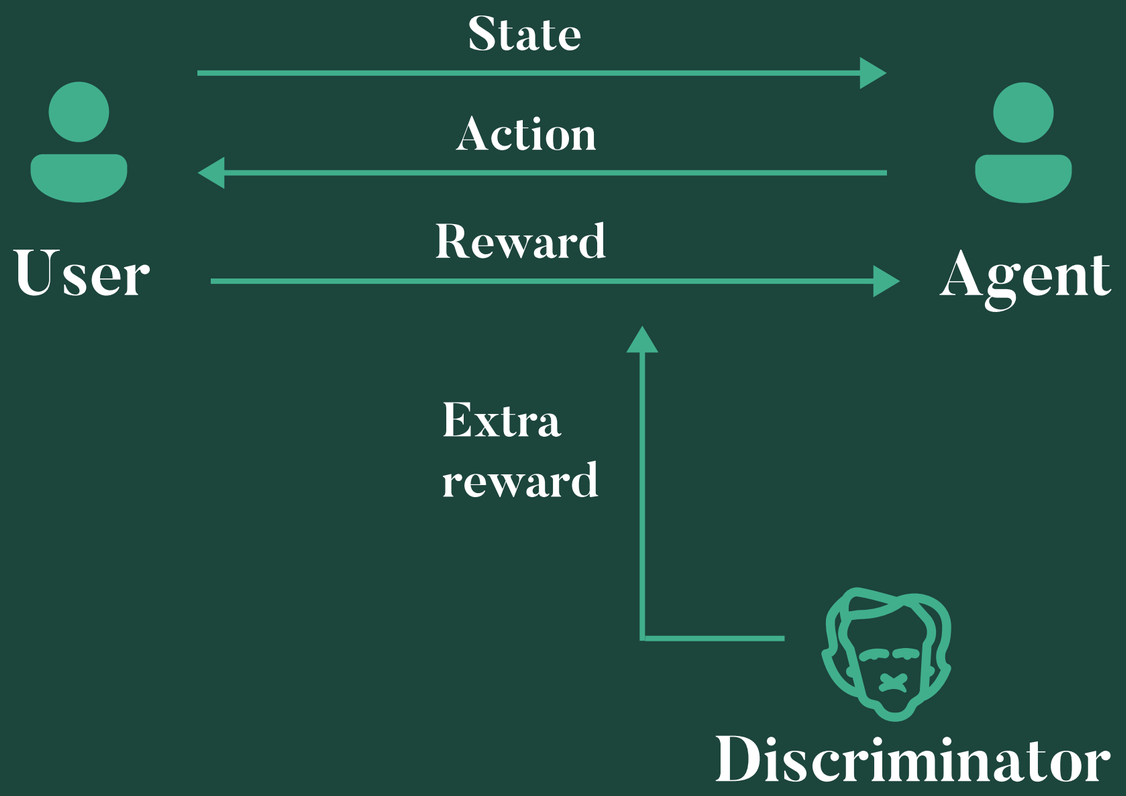
Related work

Hierarchical approach

Cheng Zhong, et al. (2022)

Related work

Adversarial approach



Ziming Li (2020)

Problem Formulation

Dataset - patient data

'explicit_inform_slots' : {'cough':True, 'vomit': True, 'fever': False}

'implicit_inform_slots' : {'headache':True, 'runny nose'': False}

'disease_tag' : pneumonia

State space

[0: 'cough', 1: 'vomit', 2: 'headache', 3:'runny nose', 4: 'anorxia', 5: 'expectoration', 6: 'Fever', ...]

state = [3, 3, 0, 0, 0, 0, 2, ...]

Action space

Action space = symptoms_set U disease_set

Action = 'runny_nose' (3)

new_state = [3, 3, 0, 2, 0, 0, 2, ...]

[3, 3, 0, 0, 0, 0, 2, ...]

3

[3, 3, 0, 2, 0, 0, 2, ...]

3: True
2: False
1: not-mentioned

PPO Agent

Trajectory: state, action, old probability, value, reward, and done array

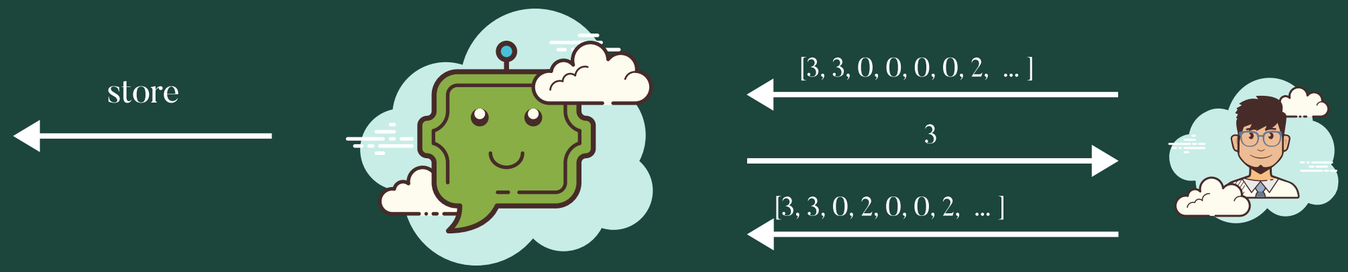
Memory

state, action, old probability, value, reward, and done array

state, action, old probability, value, reward, and done array

.
.
.

state, action, old probability, value, reward, and done array



PPO Agent

Learn()

1. Advantage Calculation
   2. For mini-batches
      3. Loss Computation
         4. Actor Loss

min(policy_ration * advantages , clip(ratio, 1- e, 1 + e)*
advantages)

5. Critic Loss

MSE(Value Estimates , Rewards)

6. Compute entropy
7. Total Loss

actor_loss + vf_coeff * critic_loss - entropy_coeff * entropy

8. Backpropagation

Auto Encoder

It consists of two main parts: an encoder and a decoder.

Encoder:

- Transforms the input data into a lower-dimensional representation (latent space).
- Captures the essential features of the input data.

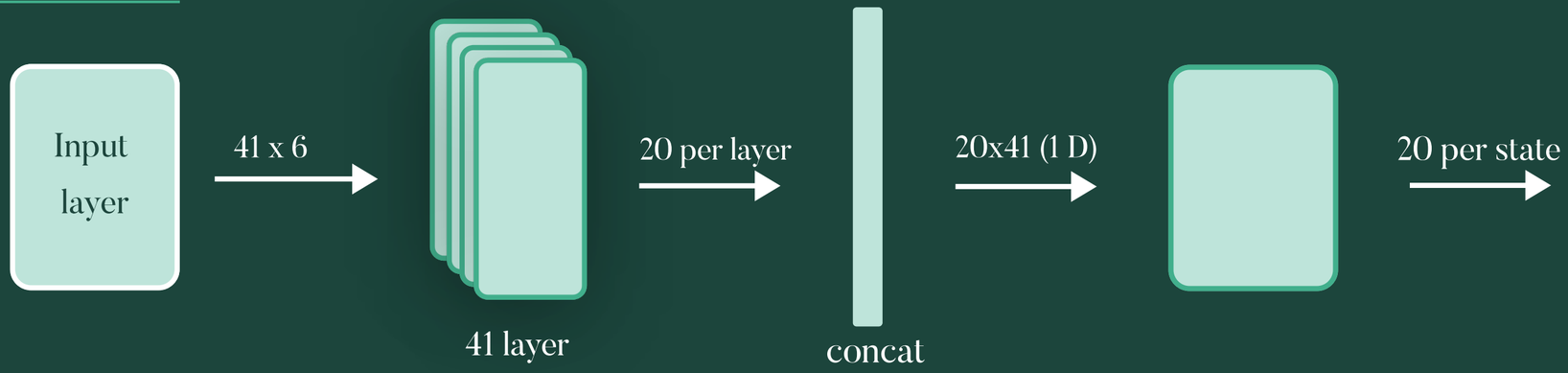
Decoder:

- Reconstructs the input data from the lower-dimensional representation.
- Aims to make the output as close as possible to the original input.

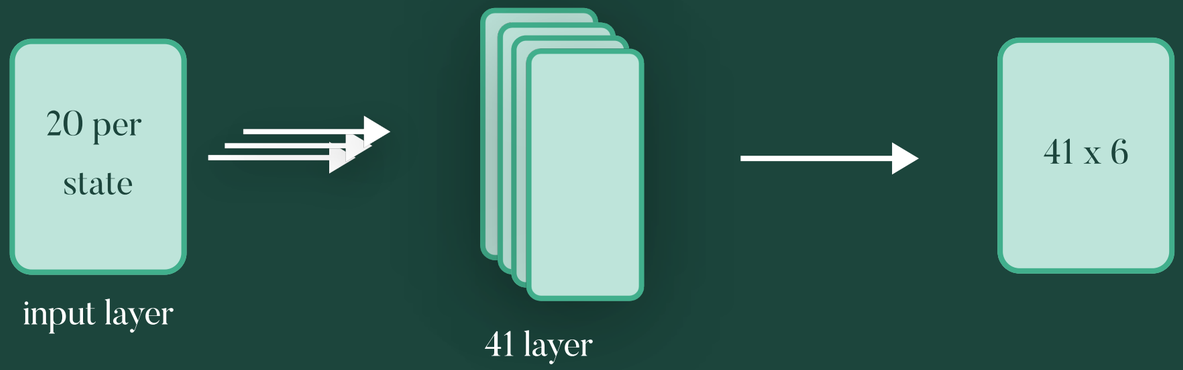
Minimize the difference between the input data and the reconstructed output.

Architecture

Encoder



Decoder



Motivation

1. Tackles categorical data issues

Proximal Policy Optimization (PPO) benefits from continuous action spaces as they allow for smooth policy updates

2. Robust to hyperparameter

PPO algorithm's performance is highly sensitive to its hyperparameters, through extensive experiments, we showed that AutoEncoder is robust to hyperparameters

3. Reward sparsity

State is not informative enough, and the reward the agent receive is sparse

Typical solutions and Experimental results

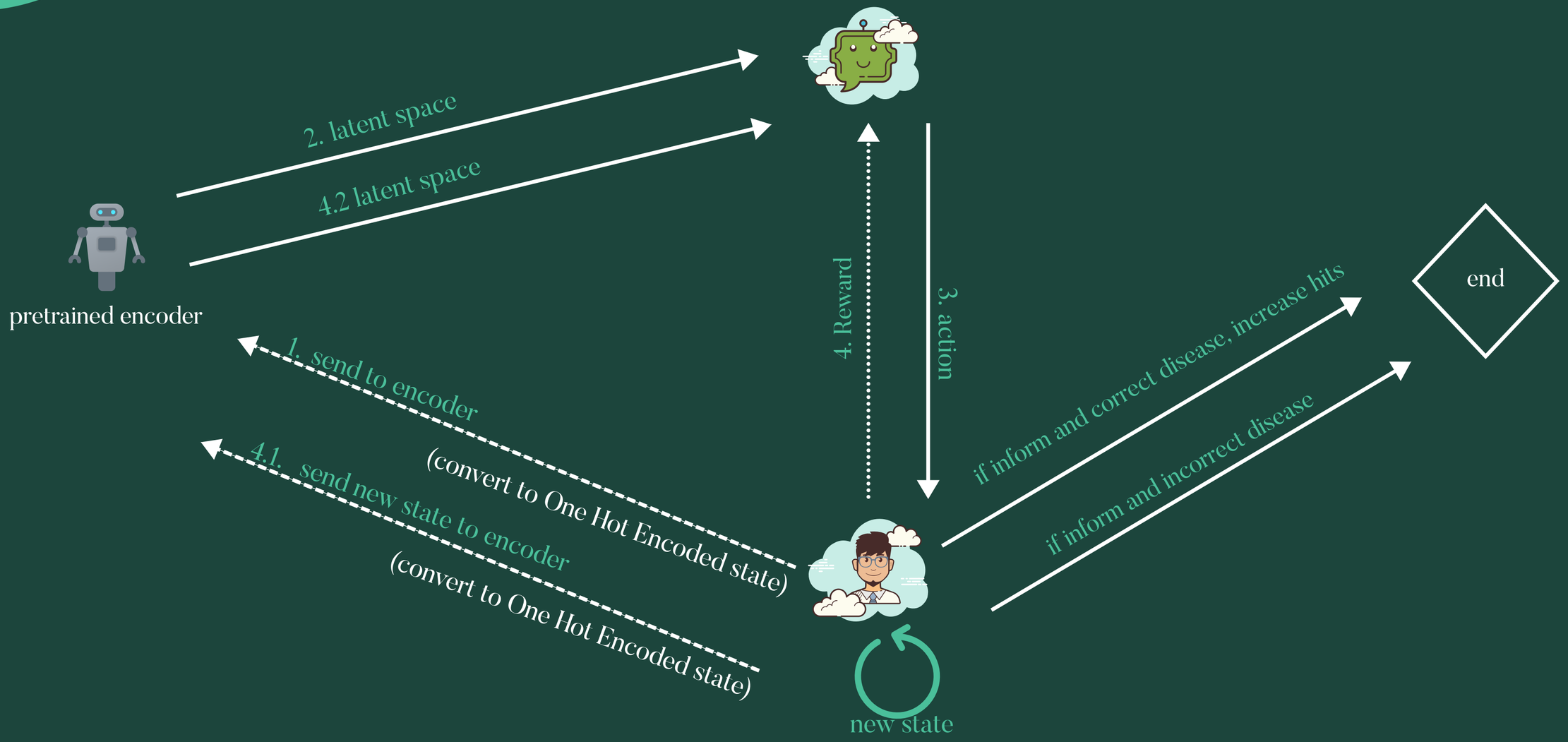
PPO One Hot Encoded Accuracy: ~ 33%

Will be discussed later

PPO with State Tracker Accuracy: ~45%

PPO with Extra Reward Accuracy: ~35%

Proposed Framework



Evaluation

We evaluate by calculating the hits the agent gets when he informs for a disease.

The dataset is already divided with 104 episodes(dialogues) for testing and 422 for training.

We calculate accuracy by dividing the number of hits over the number of episodes

Accuracy = #of hits ÷ #of episodes

Evaluation

We put the following research questions:

1. How is the performance of proposed approach compared to Baseline PPO?

2. What is the effect of hyperparameter tuning on both implementations?

3. What is the effect of different parameters on both implementations?

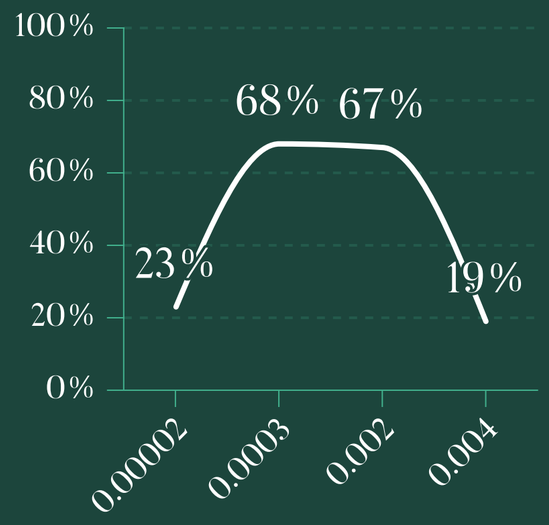
Results & Discussion

Implementation	Accuracy
PPO Baseline w/o hyperparameter tuning	~58%
PPO Baseline w hyperparameter tuning	~68%
PPO AutoEncoder w/o hyperparameter tuning	~72%
PPO AutoEncoder w Hyperparameter	**~81%**

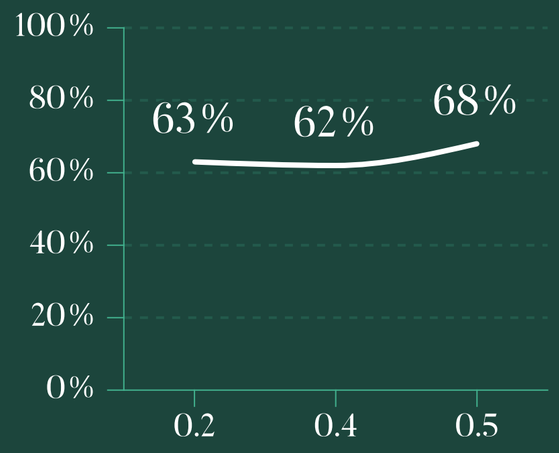
Results & Discussion

PPO Baseline

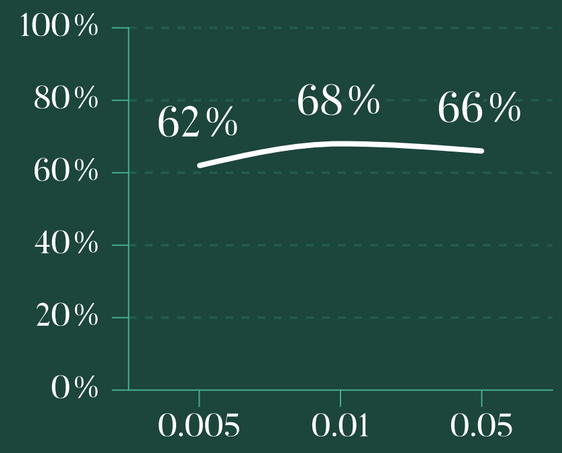
Alpha changes



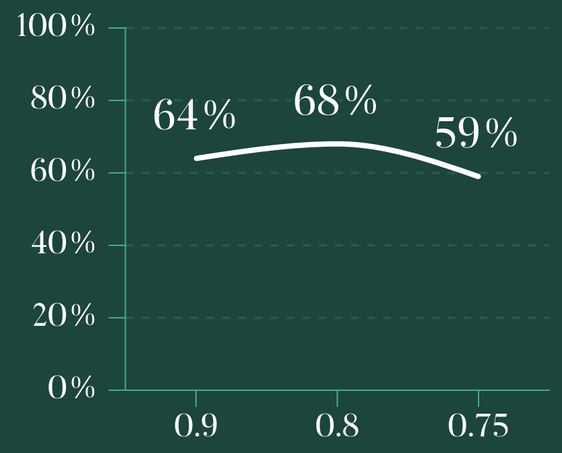
vf_coeff



Entropy_coeff



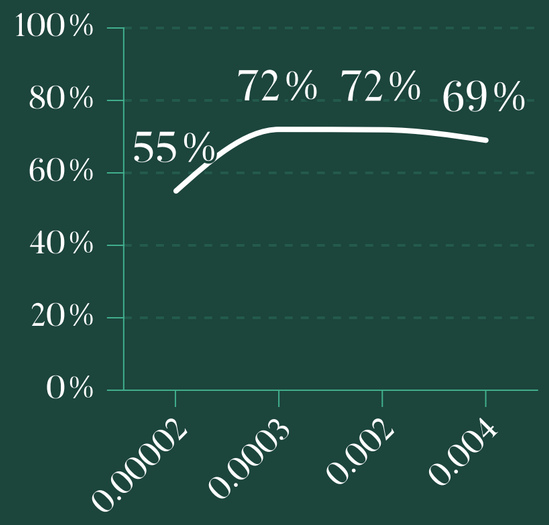
GAE Lambda



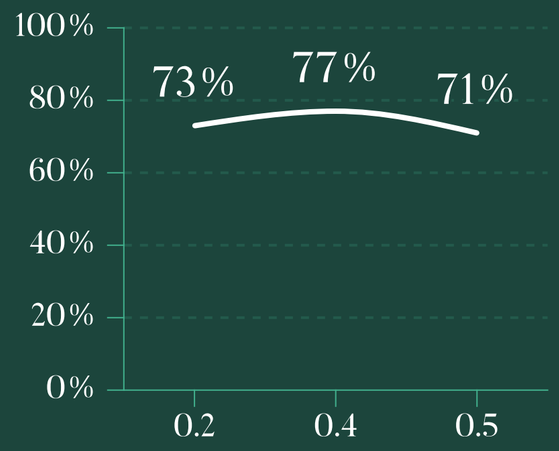
Results & Discussion

Our proposed framework

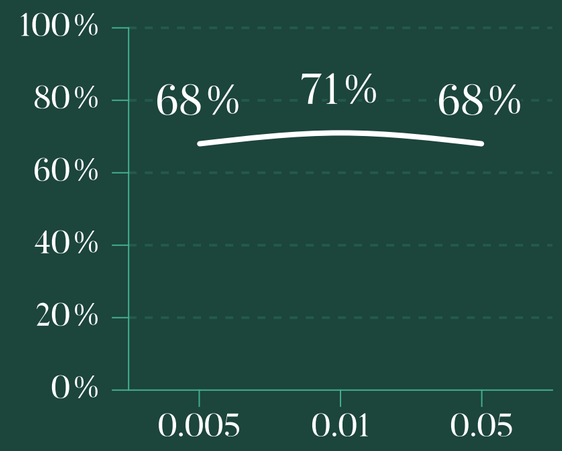
Alpha changes



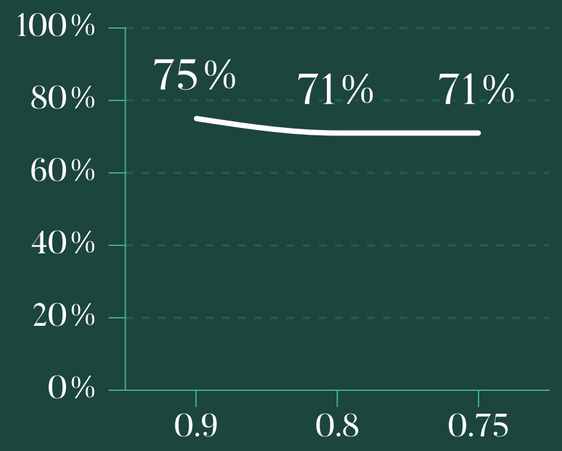
vf_coeff



Entropy_coeff

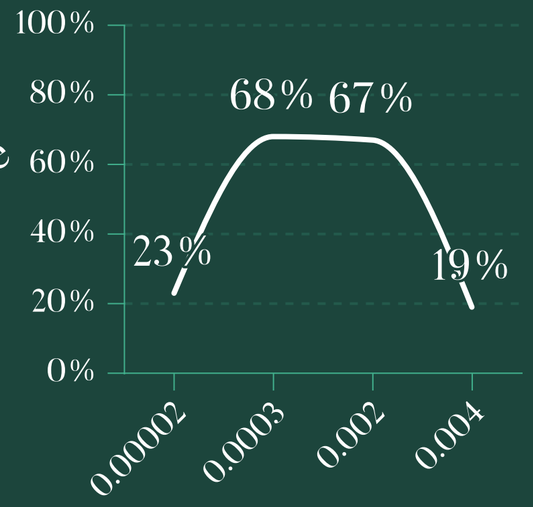


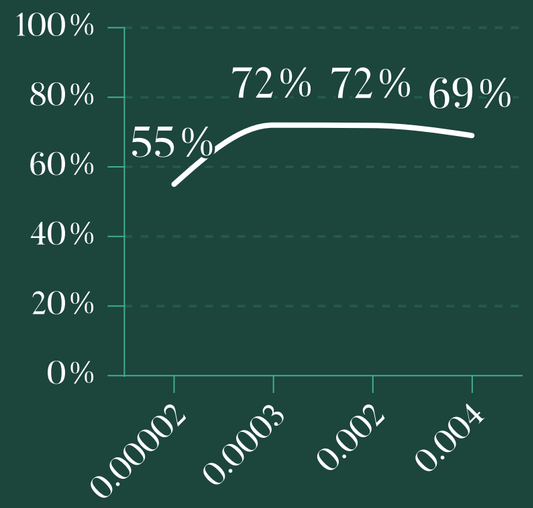
GAE Lambda



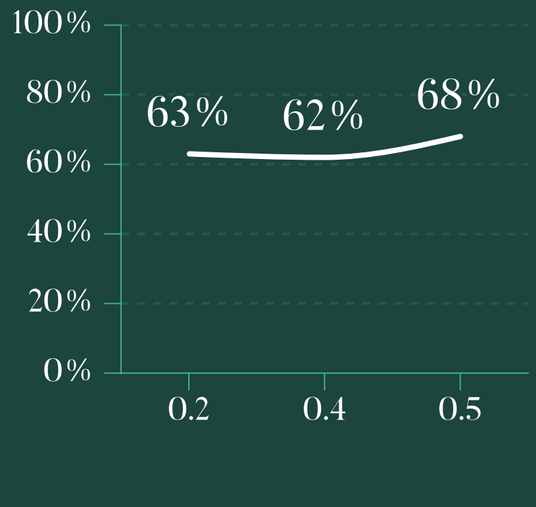
Results & Discussion

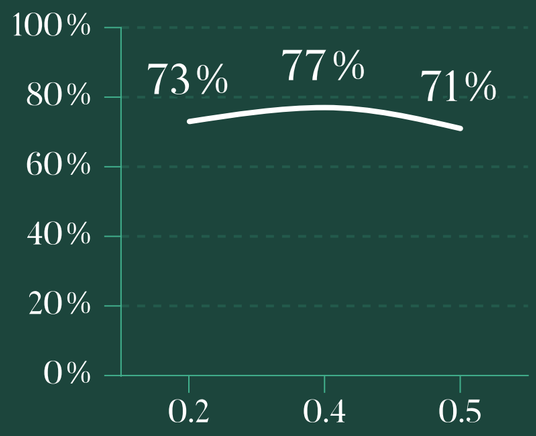
Alpha changes



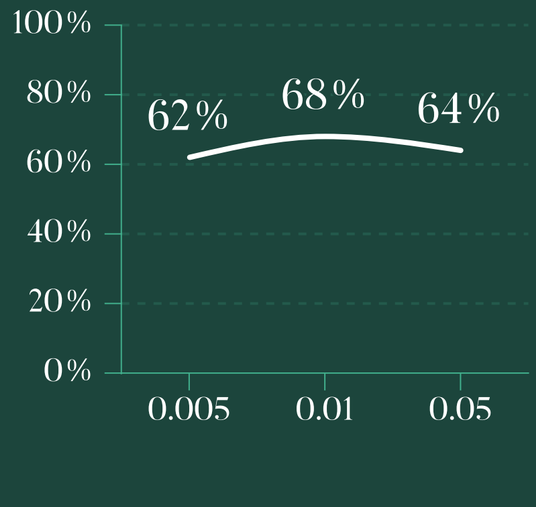


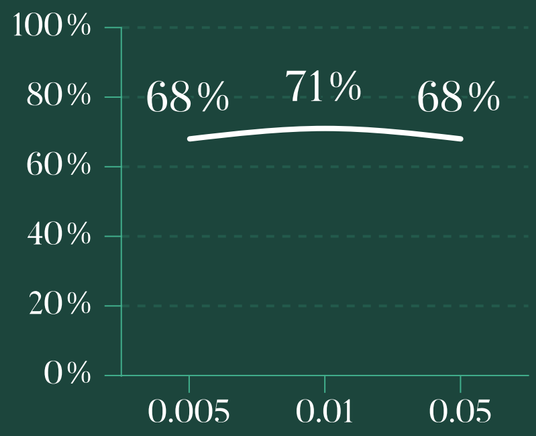
vf_coeff



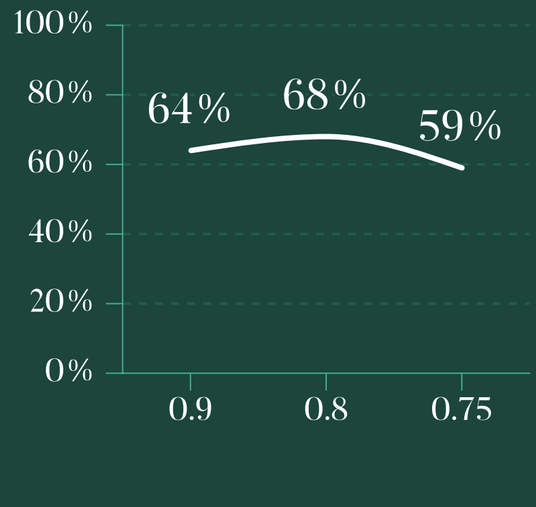


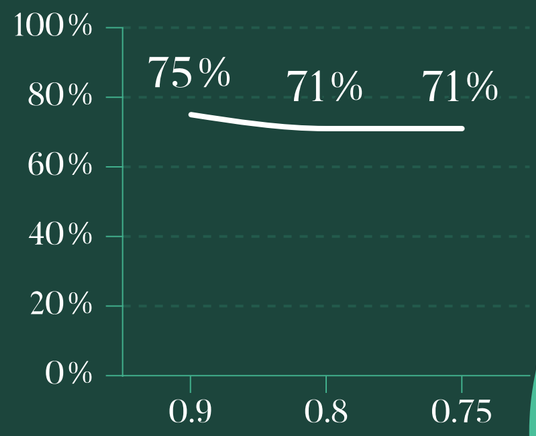
Entropy_coeff





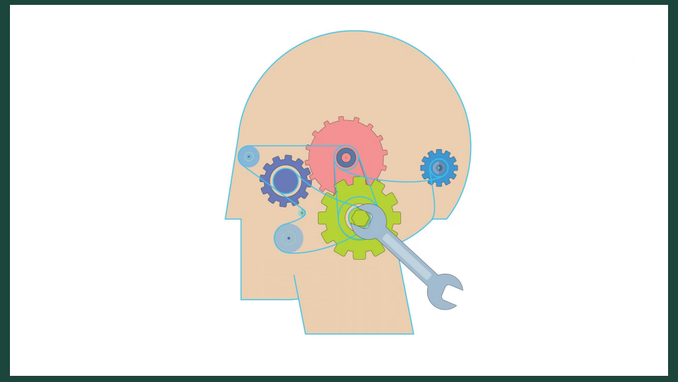
GAE Lambda





fine-tuning

The following part discuss the fine-tuning prosses and how it used in this research

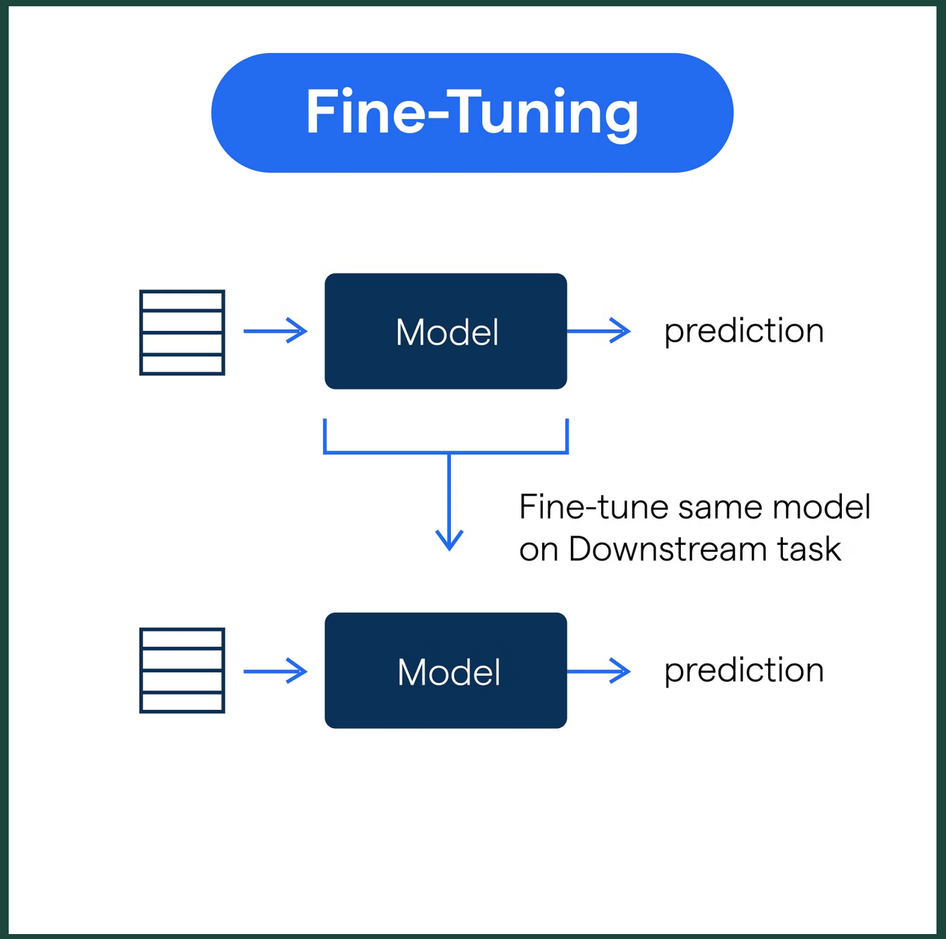


Fine-tuning

Fine-tuning generally refers to the process of taking a pre-trained model and further training it on a specific task or dataset to improve its performance for that particular task

Benefit

This technique leverages the knowledge the model has already acquired during its initial training on a large and diverse dataset. Fine-tuning is especially popular in natural language processing (NLP)



HOW does this related to our resarch

We decided to fine tune chat GPT on the same data we use on the agent that we designed and the model will deal with this data as dialogue and then give the expected disease , and after that we tested this fine-tuned model on part from this data (test data) to know how much chat GPT is better or worse than our agent and to know that we compare the accuracy for the fine-tuned model and our agent

Prerequisites

01

Pre-trained Model

A pre-trained version of ChatGPT (the version that used in this research is GPT 3.5-125 turbo)

02

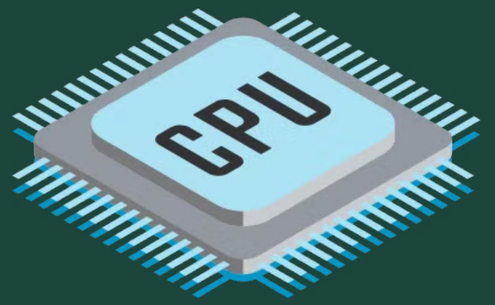
Dataset

A dataset relevant to the specific task or domain want to fine-tune the model on (the DXY dataset)

03

Computational Resources

GPUs or TPUs for efficient training (open ai API resource and Colab resource)

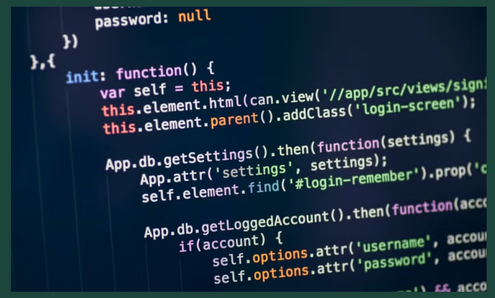


fine-tuning steps

Prepare the Dataset

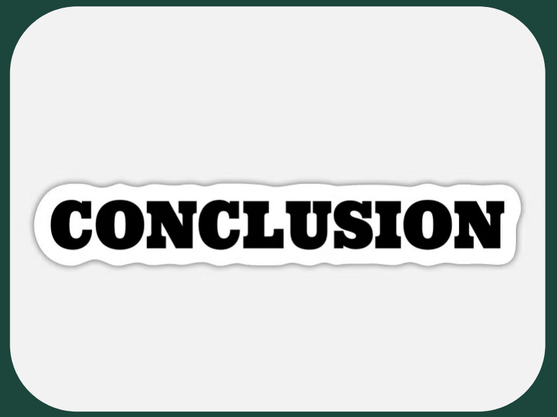
Set Up the Environment

Fine-Tune the Model



Using the model

Evaluate the Model



Conclusion of the process

1.Prepare the Dataset

a .**Collect the data:** the data used here is medical data and the dataset that used in this research is DXY(520 dialogue) dataset and this data exists in the form of request slots and each slot contain implicit and explicit symptoms and at the end the disease tag.

b.preprocess the dataset:the data should be in dialogue format as user and model have real conversation


```
{
    "request_slots": {
        "disease": "UNK"
    },
    "implicit_inform_slots": {
        "sneeze": false,
        "allergy": true
    },
    "explicit_inform_slots": {
        "cough": true,
        "runny nose": true
    },
    "disease_tag": "allergic rhinitis",
    "consult_id": "train-0"
},
```



```
{"dialogue": ["User: I have cough and runny nose.",
 "Model: Have you been sneeze?",
 "User: No",
"Model: Have you been allergy?",
"User: Yes"],
"true_disease": "allergic rhinitis"}
```

c. **Split the dataset into training**, validation, and test sets:data split in two groups one for training and the one for testing, 70% testing 30% training and evaluation that means 370 dialogue for training and 158 for testing.

training data: fit data in GPT 3.5 FORMAT to fine-tune (both format in jsonl)

```
{"dialogue": ["User: I have cough and runny nose.",
 "Model: Have you been sneeze?",
 "User: No",
"Model: Have you been allergy?",
"User: Yes"],
"true_disease": "allergic rhinitis"}
```

```
{"messages":[{"role":"user","content":"I have cough and runny nose."},
{"role":"assistant","content":"Have you been sneeze?"},
{"role":"user","content":"No"},
{"role":"assistant","content":"Have you been allergy?"},
{"role":"user","content":"Yes"},
{"role":"assistant","content":"true_disease: allergic rhinitis"}]}
```

testing data:is preprocessed by making the data as dialogue and splitting the data in two groups, one for conversation and one saving just the true diseases.

```
User: I have Diarrhea and Restlessness and weight loss.

User: I have Diarrhea and anorexia and rash.
Model: Have you experienced Oliguria?
User: Yes
Model: Have you experienced weight loss?
User: Yes

User: I have Herpes and rash.

User: I have cough and sneeze and runny nose.
Model: Have you experienced Vomit?
User: Yes
Model: Have you experienced allergy?
User: Yes
```

dialogue

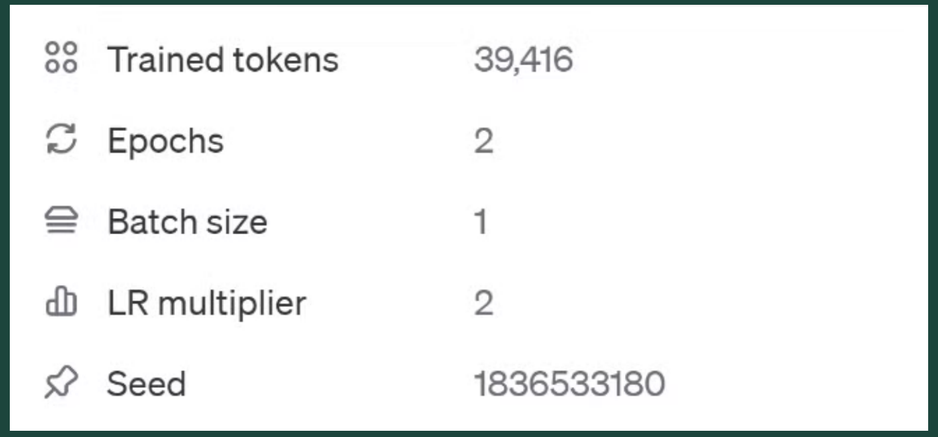
```
Pediatric Diarrhea
Pediatric Diarrhea
Children hand-foot-mouth disease
allergic rhinitis
Pediatric Diarrhea
```

true diseases

2.Set Up the Environment: Colab used as workspace and the UI open ai API also as tool for fine-tuning.

3.Fine-Tune the Model:

Trained Tokens: The total number of tokens (words or subwords) used during the fine-tuning process.
Epochs: **Definition**: The number of complete passes through the entire training dataset.
Batch Size: **Definition**: The number of training examples of the model processes before updating its internal parameters.
LR Multiplier (Learning Rate Multiplier): **Definition**: A factor that scales the base learning rate used during training.



Does fine tuning Gpt 3.5 turbo Relly affect the model & it’s output?

Subtle Adjustments:the fine-tuning data slightly influences how the model prioritizes certain responses or patterns.

Contextual Relevance: If the prompt closely matches scenarios in the fine-tuning data, the influence of the fine-tuning will be more noticeable

yes!

4.run the new model:

In this point the model used throw using the open **API key** to have access to the models and the **name of the fine-tuned model** to get the access of this model, by giving the split data dialogues (not the true disease) then the model gives what is the expected diseases of each dialogue

5.Evaluate the Model:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

we got 54,43% accuracy, that means we got 86 correct predictions out of 158 (the total number of the predictions)

Conclusion of the process

fine-tuning GPT-3.5 Turbo, users can enhance the model's performance on targeted applications, ensuring more accurate, contextually appropriate, and relevant responses. This process involves adjusting the model based on custom datasets, which can include extensive context, detailed instructions, and comprehensive examples

note: this process costs around 10 $

so, this process shows us that our agent is doing well and gets high accuracy in dialogue medical diagnosis on this data.

References

1. https://medicalxpress.com/news/2023-10-ai-early-pancreatic-cancer.html
2. https://news.mit.edu/2024/new-hope-early-pancreatic-cancer-intervention-ai-based-risk-prediction-0118
3. https://www.echelon.health/the-role-of-ai-in-early-disease-detection/
4. https://www.mckinsey.com/industries/healthcare/our-insights/tackling-healthcares-biggest-burdens-with-generative-ai
5. Hierarchical Reinforcement Learning for Automatic Disease Diagnosis, Cheng Zhong, et al. (2022)
6. Guided Dialogue Policy Learning without Adversarial Learning in the Loop , Ziming Li

Names:	Student ID	Email
Mohammad Al Tamimi	144754	mmaltamimi20@cit.just.edu.jo
Abdelrahman Othman	147042	adothman20@cit.just.edu.jo

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)