Fine-tuning

**Introduction**:

Fine-tuning a pre-trained language model like ChatGPT involves adapting it to specific tasks or domains by training it further on a more focused dataset.

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).

This process enhances the model's performance in specialized applications, making it more useful for particular use cases.

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**

1. **Pre-trained Model**: A pre-trained version of ChatGPT (the version that used in this research is GPT 3.5-125 turbo)
2. **Dataset**: A dataset relevant to the specific task or domain want to fine-tune the model on (the DXY dataset)
3. **Computational Resources**: GPUs or TPUs for efficient training (open ai API resource and Colab resource)

here bellow we discuss the steps for doing this prosses and giving some notes:

1. **Prepare the Dataset**:

Gather and preprocess the dataset for the specific task. This involves data cleaning, normalization, and splitting into training, validation, and test sets.

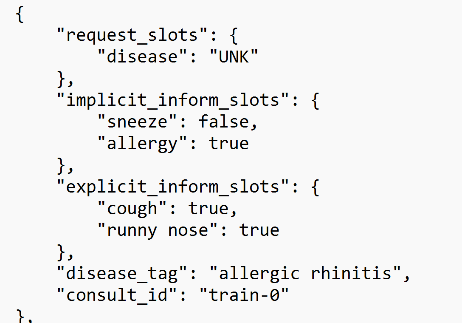
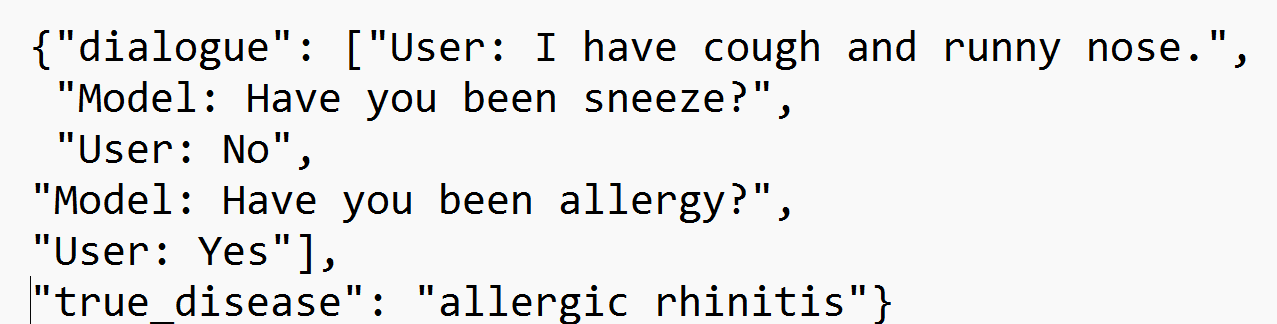
a .**Collect the data**:

Preparing dataset is the most important thing in fine-tuning, the data used here is medical data and the dataset that used in this research is DXY 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** and ensure the data is cleaned and formatted correctly

to fine-tune this data, **first** the data should be in dialogue format as user and model have real conversation, so at first the user will start the saying the explicit symptoms then the model will ask about the implicit symptoms one by one and the user will answer yes or no at the end the model will give the diagnose

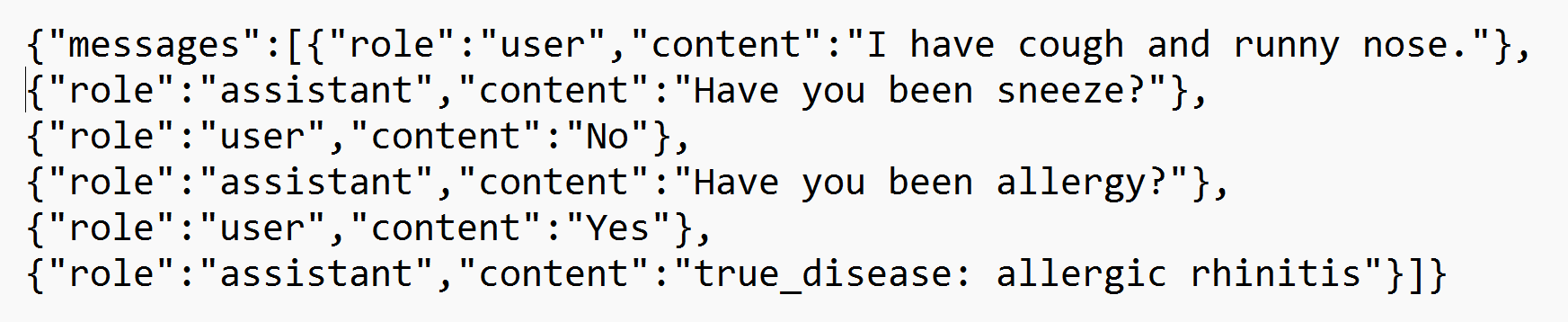
ex:



c. **Split the dataset into training**, validation, and test sets.

And data split in two groups one for training and the one for testing, 70% testing 30% training and evaluation.

Then **training** data need to be fit in format that GPT 3.5 -125 turbo to accept and understand , each dialogue will be understand as massage to the model and the the user and model will be roles in this massage (dialogue) and the questions and answers will be the content of what the roles saying, and this data converted to Jsonl format so in this way model can understand the each line as object (Json).  
  
ex:

A white background with black text

Description automatically generated

Also, the 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.

1. **Set Up the Environment**:

Colab used as workspace and the UI open ai API also as tool for fine-tuning.

1. **Fine-Tune the Model**:

* Train the model on the task-specific dataset. This involves adjusting hyperparameters like learning rate and batch size to prevent overfitting.
* Techniques such as learning rate scheduling and early stopping can be used to optimize the fine-tuning process.

**the training data used as prompt to fine-tune the model and the model has limit 64,000 tokens for the training data and the below parameter manage the way the model will learn on this data :**



Here's an explanation of each parameter displayed in the image:

**Trained Tokens**:

* + **Definition**: The total number of tokens (words or subwords) used during the fine-tuning process.
  + **Significance**: Indicates the volume of data the model has been trained on. More tokens generally allow the model to learn more patterns and nuances from the training data.

**Epochs**:

* **Definition**: The number of complete passes through the entire training dataset.
* **Significance**: Each epoch allows the model to learn from the entire dataset. Multiple epochs help the model to refine its understanding and improve performance. However, too many epochs can lead to overfitting, where the model performs well on training data but poorly on unseen data.

**Batch Size**:

* **Definition**: The number of training examples of the model processes before updating its internal parameters.
* **Significance**: Smaller batch sizes (like 1) can lead to more precise updates and potentially better generalization, but they can also increase training time. Larger batch sizes speed up training but may require more memory and could lead to less precise updates.

**LR Multiplier** (Learning Rate Multiplier):

* **Definition**: A factor that scales the base learning rate used during training.
* **Significance**: Adjusting the learning rate helps control how much the model's weights are adjusted with each update. A higher multiplier increases the learning rate, allowing the model to learn faster, while a lower multiplier makes learning more gradual and stable.

**Seed**:

* **Definition**: A value used to initialize the random number generator for the training process.
* **Significance**: Ensures reproducibility of the training process. Using the same seed value allows for the exact replication of results, which is crucial for comparing experiments and debugging.

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

fine-tuning GPT-3.5 Turbo can significantly affect the model by enhancing its performance on specific tasks and tailoring its outputs to better align with the requirements of a particular application. Here's a deeper look at how fine-tuning impacts the model and have good training:

* **Focused Training**: The token limitation necessitates a highly focused and efficient training process. The fine-tuning dataset must be carefully curated to include only the most relevant examples, reducing redundancy and ensuring that each token contributes to learning useful patterns.
* **Iteration and Optimization**: Fine-tuning within a limited token budget requires iterative training and evaluation to optimize the model’s performance. This often involves multiple rounds of training with progressively refined datasets.

### **Understanding the Impact of Fine-Tuning on a Small Dataset**

**Subtle Adjustments**: Fine-tuning on a small dataset makes subtle adjustments to the model’s weights. This means that while the model's overall knowledge remains largely based on the original pre-training, the fine-tuning data slightly influences how the model prioritizes certain responses or patterns.

**Contextual Relevance**: When generating responses, the model considers the entire context of the input prompt. If the prompt closely matches scenarios in the fine-tuning data, the influence of the fine-tuning will be more noticeable. Otherwise, the response will primarily reflect the general knowledge and patterns learned during pre-training.

**Fine tune cost: fine** tuning model could be free or paid it depends on the model wants to be fune-tuned ,fine-tuning GPT 3.5 turbo is paid and this fine-tuning cost 10$ for training and testing the model (for each 100,000 tokens take 2.5$) and fine-tuning has been applied many times on the model to see the different results and what is the good data form.

1. **Using the 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.**

**These expectations considered the result of our new model.**

1. **Evaluate the Model**:

Evaluating the model performance is very various and there are many principles of it, but here we focus on the accuracy of the model results (expectations),

The accuracy of the model calculated as follow:

A black text on a white background

Description automatically generated

We put counter to know how many diseases model expected truly, and the divided on the total number of the predictions.

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

1. **Conclusion of the process**:

Fine-tuning GPT-3.5 Turbo offers significant advantages for tailoring the model to specific tasks or domains. This expanded capacity allows for the incorporation of more complex and detailed training data, facilitating the creation of more specialized and effective models. By 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  
so, this process shows us that our agent is doing well and gets high accuracy in dialogue medical diagnosis on this data.

**Dedication**

**To our dear doctor, who helped us and followed us and served as a guide for us in our project and a was like compass for us Dr.Qanita bani Baker**

**at the end, we dedicate this research that was presented to the martyrs, heroes, and children of Gaza who sacrificed everything they had, and we dedicate this research to their pure souls.**

**May allah use us to support His religion.**

Here below the links for codes that wrote to do this prosses:

Data preprocess  
  
 the first step to fine tune any moudel is preparing data , so first we change the text dataset (the real one) to json format second we change the json format data to dialoge (user and module) third we export the dialoge as jsonl format

1. <https://colab.research.google.com/drive/1cY59Han3JDZR1knxtlhZowCPU6YHb-Qm>

step 2:from dialode text to chat gpt formatt

1. <https://colab.research.google.com/drive/1e0Zkjl_YZoJSasrDBQtzpkD3YuvNVSfp>

preparing testing data

step3:converting testing data from dialogue to another and split the conversation and the real disease

1. <https://colab.research.google.com/drive/1r5sEdrFCnvjN8HO2jkPtQ8woCJd0ZL-t#scrollTo=_DuVPrlHwMZg>

step4:run the fine tuned model and get output

1. <https://colab.research.google.com/drive/1szVCKFtE1hHcp9X8ySFLc8zmVSqeH80g>

splitting codes in this way make working on it easier and can make changes with no fear of dependencies