|  |  |  |
| --- | --- | --- |
|  |  | Research notes  NAZMUS SAMMO-103512692 |

Date: 21 October 2023

This will be the fortnightly last report for our project, Machine Reading Comprehension with generative AI for our case it’s BERT which is fine tuned on medical domain with Bao. It was one of most interesting and advanced topic I have done in my university 2nd year. In this report I will talk about the last fortnightly work I have done for this project. To comprehend last week’s work what I have done is, I took cpgqa dataset and fine tuned on bert based model with the technique of full fine tuning and P-tuning. For p tuning it was performing very poor but for full fine tuning for the eval set of squad it scored almost 40% and for eval set of cpgqa it scored 45% around which is not bad but not good either, there were plenty of room for improvement. On the meeting when I was speaking with Bao he said you can improve it a lot by using a bert model which is already fine tuned on squad and then fine tune it on cpgqa. As Bao said, I have done that this week, and it also make sense to me because cpgqa is comparatively a small dataset to train our model, so fine tuned on squad then on cpgqa will definitely give me an edge.

* So, for the pretrained bert model on squad I have used csrron/bert-base-uncased-squad-v1 for further process. So after taking the model what I have done first is evaluated the model on top of squad v1 and also squad v2 to see how it’s performing.

A screenshot of a computer program

Description automatically generated

* From the result we can see is that it has been fine tuned on squad very well that it got 88% f1 score and 80% of exact match which is definitely great.
* After that to see how it will perform which squad v2 just of curiosity I have evaluated on squad v2 eval set as well and my expectation was really high. Compared to the score it got with squad v1.
* A screenshot of a computer

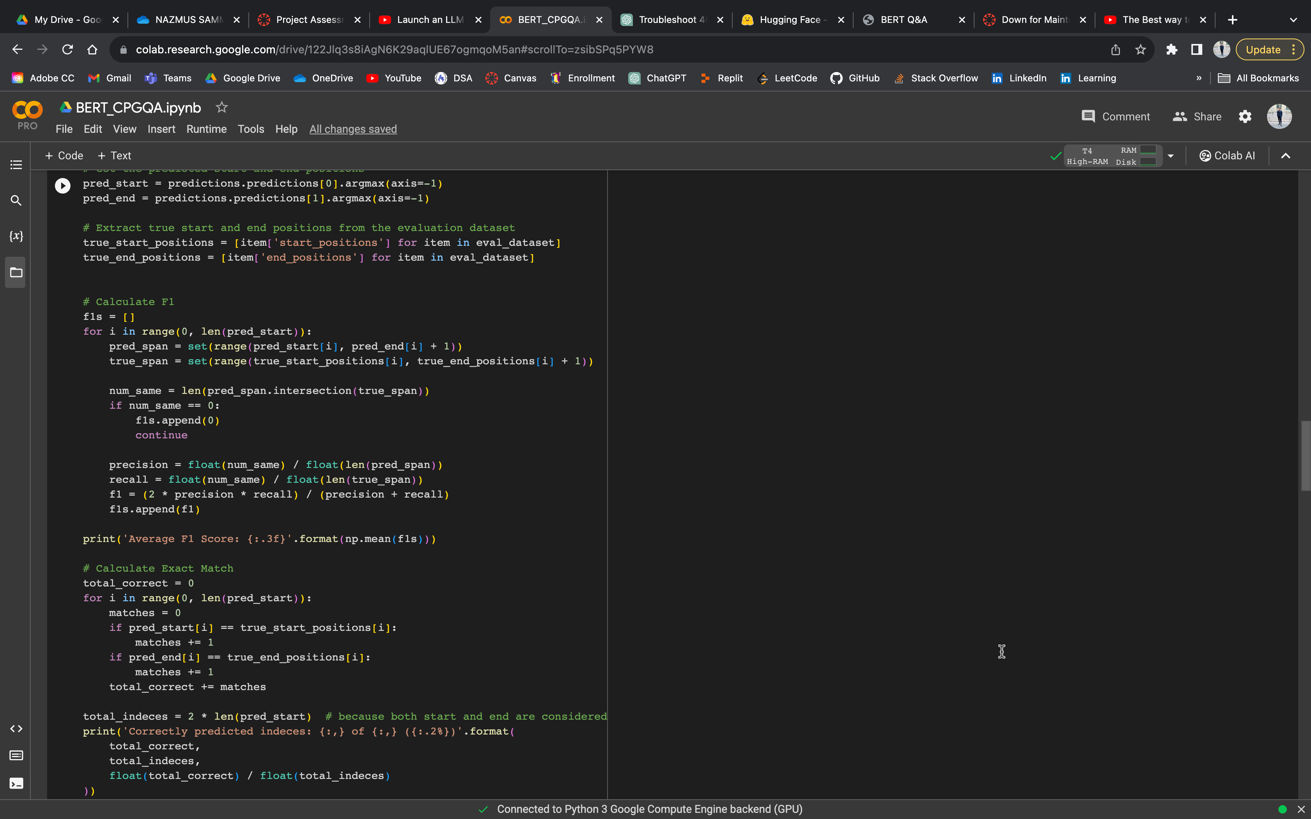
  Description automatically generated
* But the score was not that great, where it got the f1 score of 44% only but then I studied about the difference of these 2 dataset and understood the difference. The primary difference between SQuAD v1 and SQuAD v2 is that SQuAD v2 includes unanswerable questions in addition to answerable ones, while SQuAD v1 contains only answerable questions. So the evaluation result is like that.

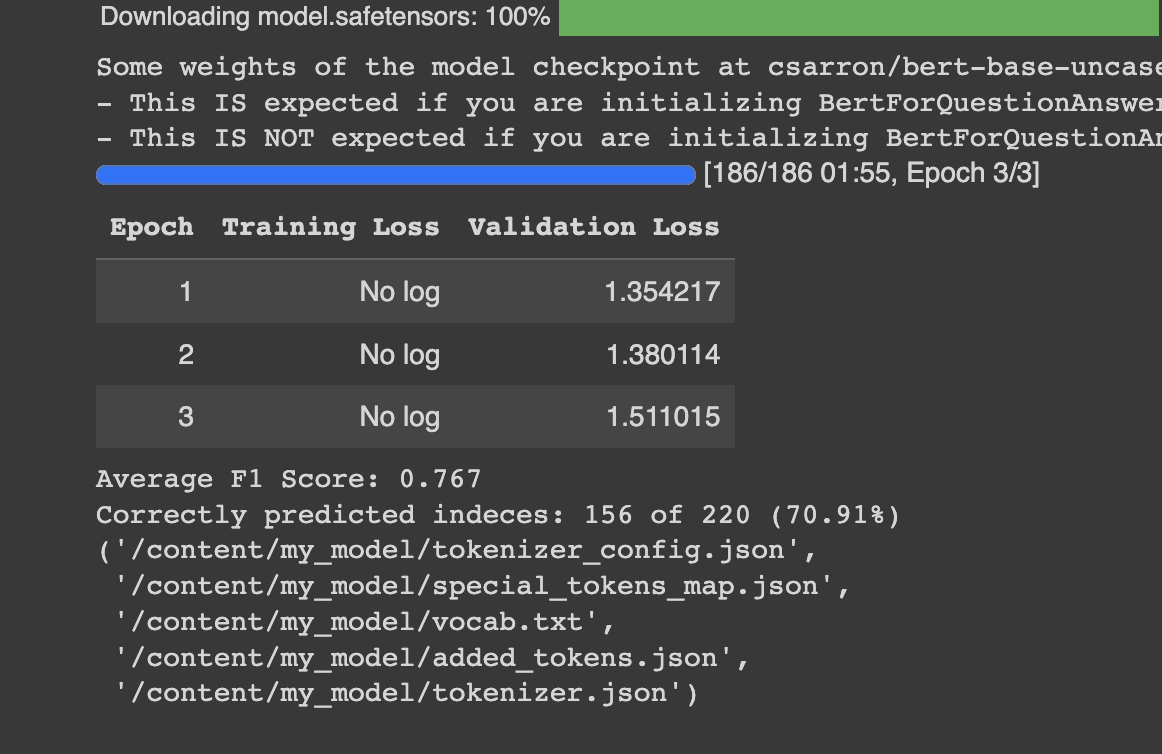
So as I have done the evaluation of the primary model now my main work is to get this model fine tuned on squad with the technique of full fine tuning, and also just to compare peft method for low parameter model like BERT I have done the P-tuning as well.

I will discuss about the full fine tuning of cpgqa on top of pretrained bert model fine tuned on squad.

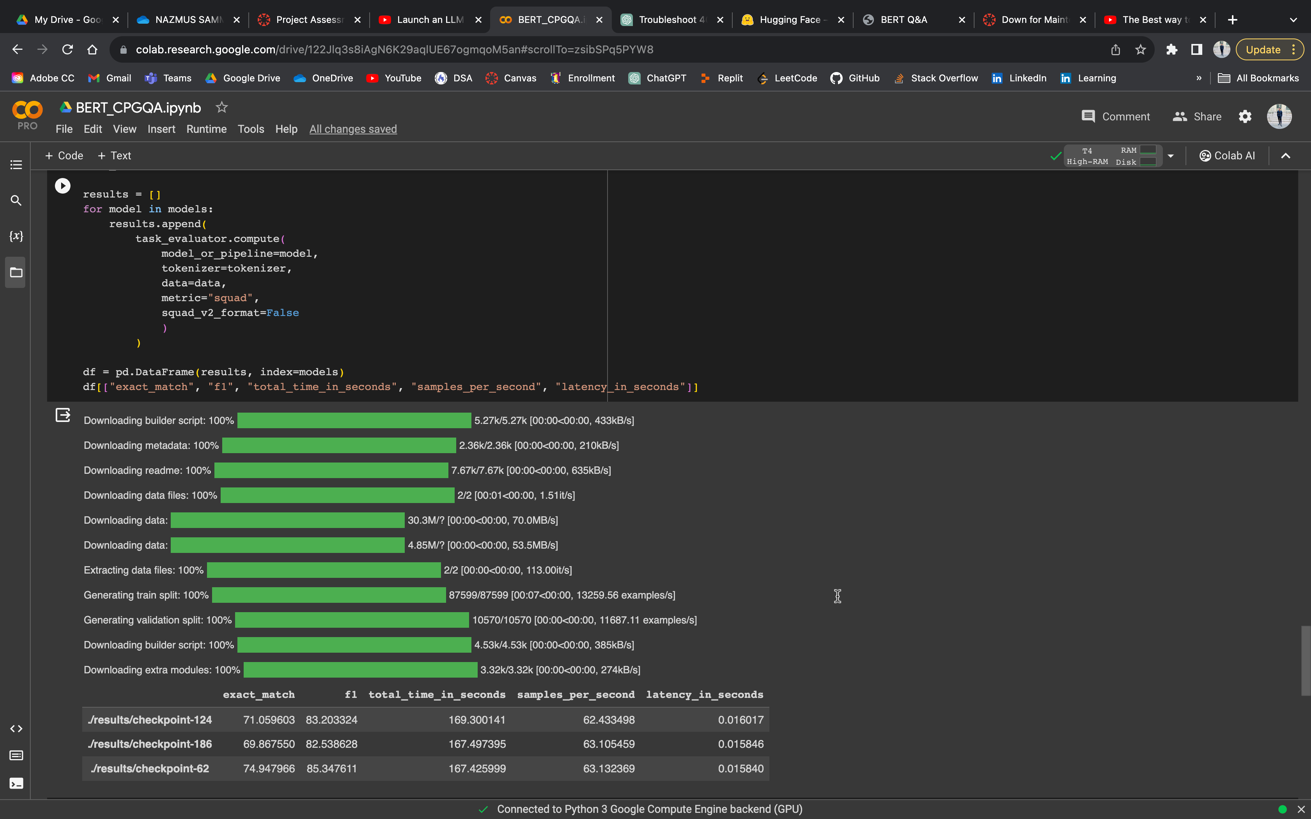
A screenshot of a computer

Description automatically generated





And the result came out as Bao said. Almost doubling the f1 score and it’s 77% which is pretty good for cpgqa dataset. And then just for testing purpose how the full tuning impacts the weights of the model and how it’s doing now with the eval set of squad v1.



From here we can see that the model is getting around 82% of score compared with before fine tuning 88%. From here we can see that the model is not facing that big of a catastrophic forgetting what we usually expect from full fine tuning. (But there’s an issue with peft p-tuning which I will discuss later.)

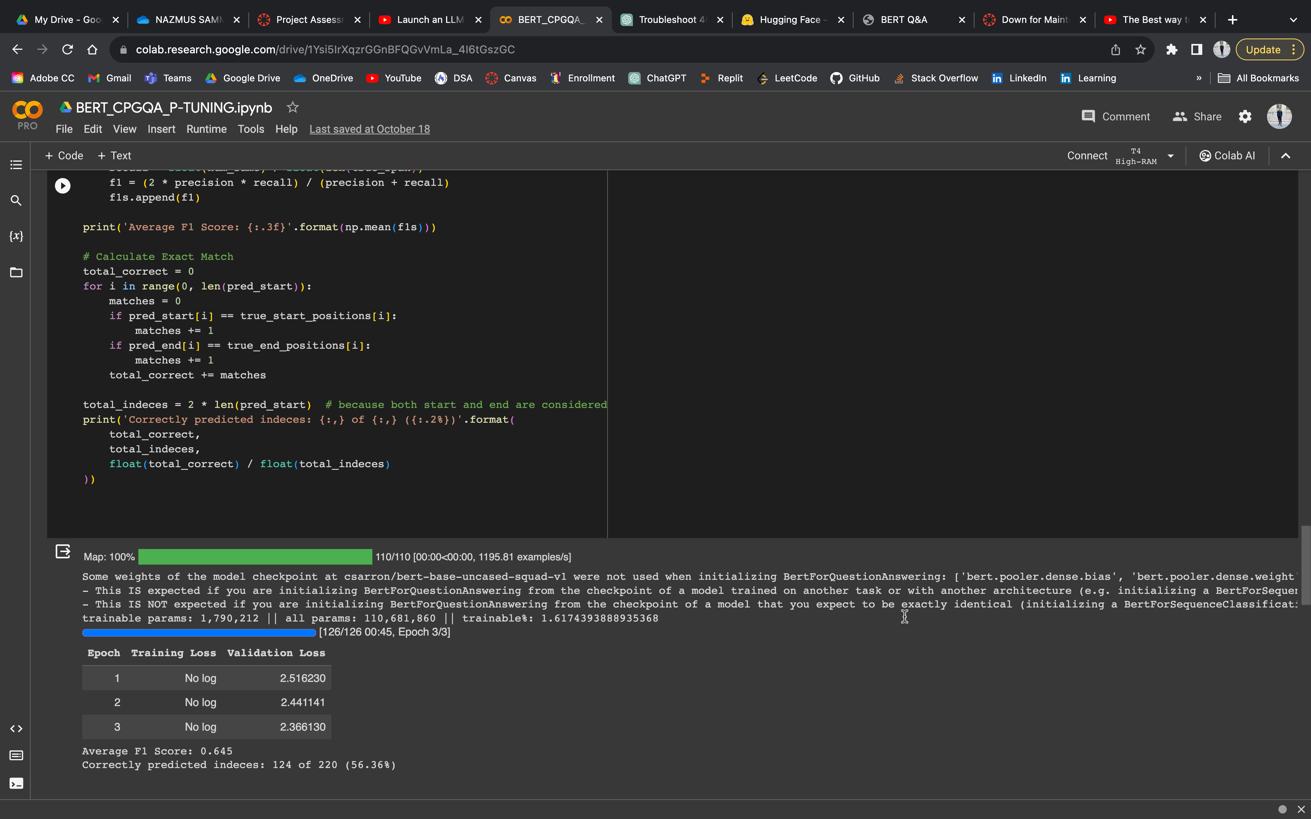
* Now as for comparison reason I have done the p-tuning on cpgqa on bert pretrained on squad v1.

The p-tuning configuration is:

A screenshot of a computer

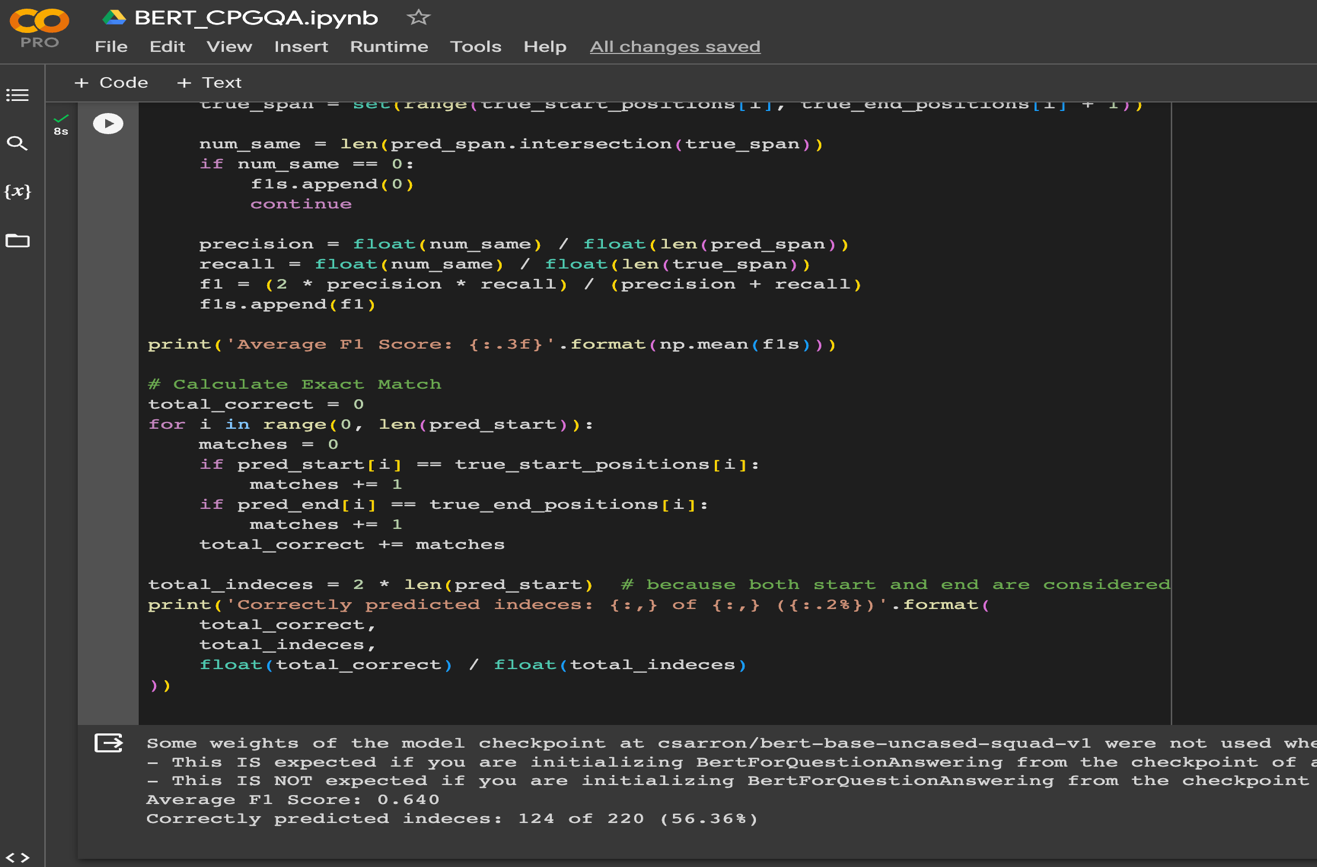
Description automatically generated

This is the same configuration I have used the previous week just make sure the comparison is fair. Just to say a bit about previous bert base p tuning score, the f1 score was bad which is only 8%. But now I have done the p-tuning on top of the bert based uncased pretrained on squad v1. And the result is:

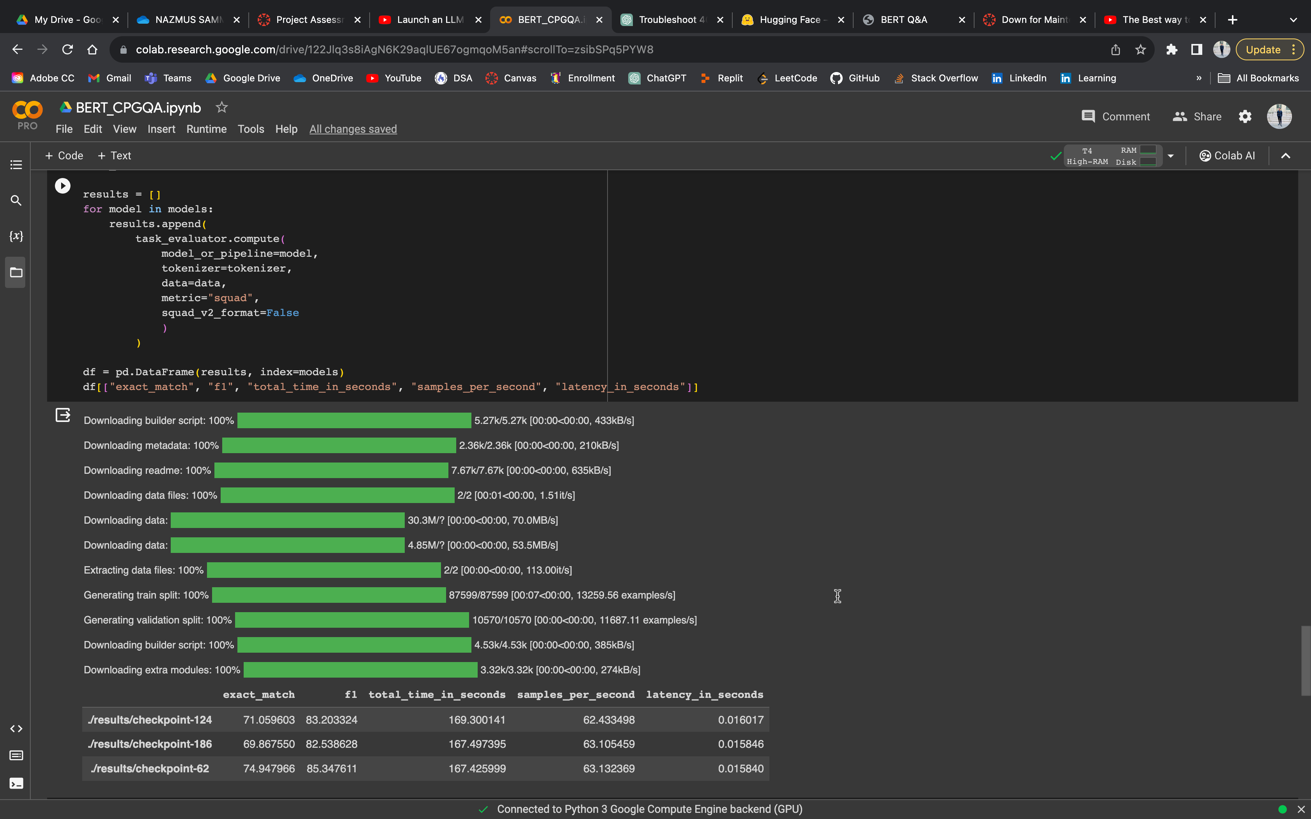
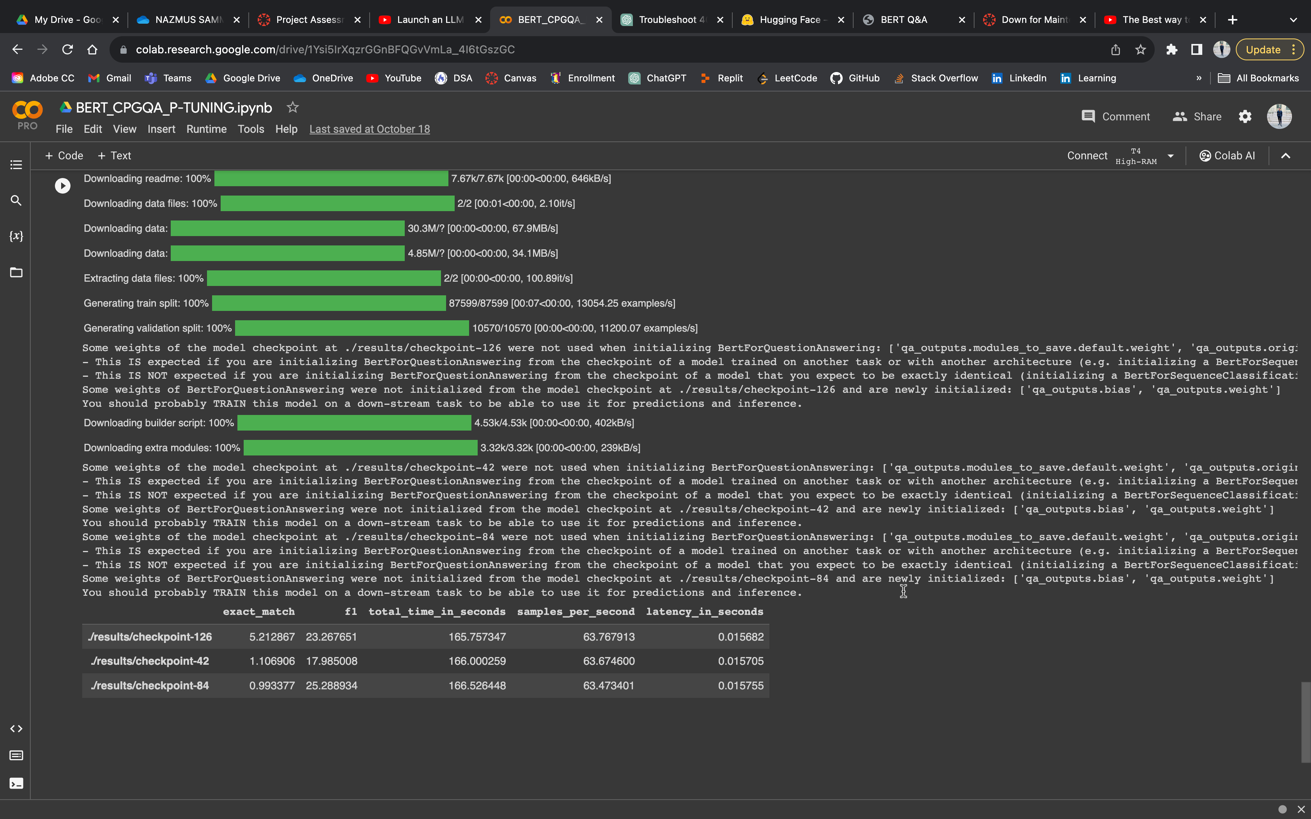


The f1 score is 64% now. From here we can that the f1 has improved drastically. But from here we can write a conclusion that it’s not the cpgqa p-tuning method which is making the model better but it’s the pretrained squad which is making the model better.

* To more strengthen the conclusion we can run a test which will be: Evaluate the pre trained bert model which is fine tuned on squad with the test split of cpgqa,

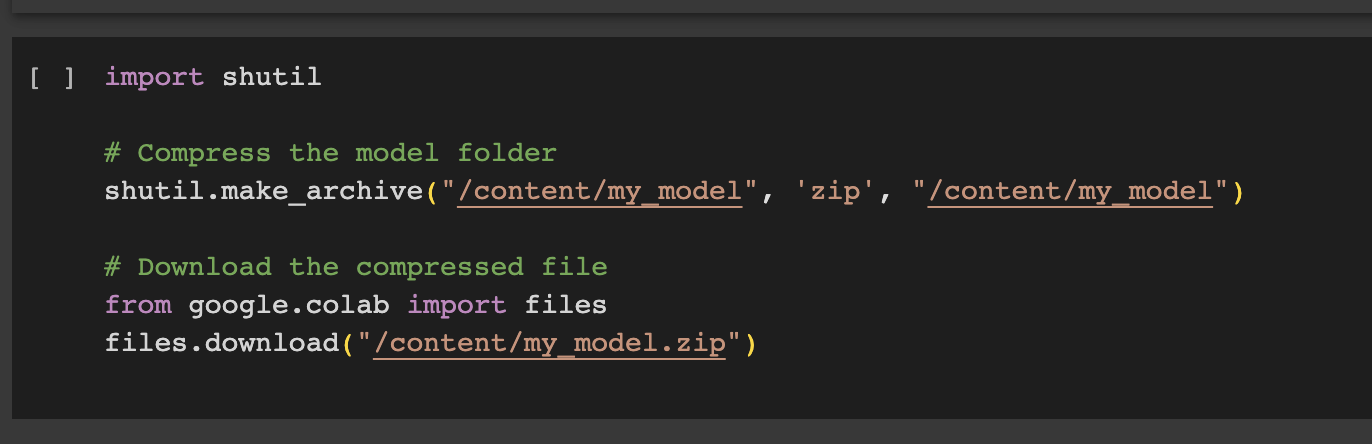


From here we can draw the conclusion that it’s not the p tuning which makes the model better but the based squad fine-tuned which performs great on the test set of cpgqa.

* **Discussion Point:**
* So as I have studied previously that full fine tuning increase the risk of catastrophic forgetting of the model and peft method decrease the risk of it.
* But from my work, if we look before full fine tuning the squad fine tuned model on cpgqa and the eval split is squad then the f1 score is 88% and after the full fine tuning the score is around 84%.
* 
* If we look before peft p-tuning the squad fine tuned model on cpgqa and the eval split is squad then the f1 score is 88% and after the peft p-tuning the score is around 24%. Which is very less and we can say it gone through a massive catastrophic forgetting which should not be the case for it. I am eager to learn about it more from you Bao. Because I can’t draw any conclusion about it because it’s against the theory.
* 

So, these are pretty much the fine tuning I have done this week.

But for this week, I was going through the the requirement of the project and one point catches my mind is that we need to Implement a UI for the user to interact with the model. And it seem to be forgotten by us. So I have a few projects about about full stack web development I wanted to make the frontend and backend for it (though if I have worked more on it I could have integrated the database with it which could capture the previous conversation and can be saved in the database and can be viewed late, but if you want I can do that as well). So what I have done is, saved the fine tuned to my device from colab.



Then I unzipped it and used the model in my backend.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

I loaded the model and make a post api for it. The api documentation is given below.

**API Endpoint**

**Endpoint**: **/predict**

**Method**: **POST**

**Request Payload**

The API expects a JSON payload with the following structure:

{ "context": "Your context here...", "question": "Your question here..." }

* **context** (string, required): The context or background information based on which the answer should be extracted.
* **question** (string, required): The question for which an answer is needed.

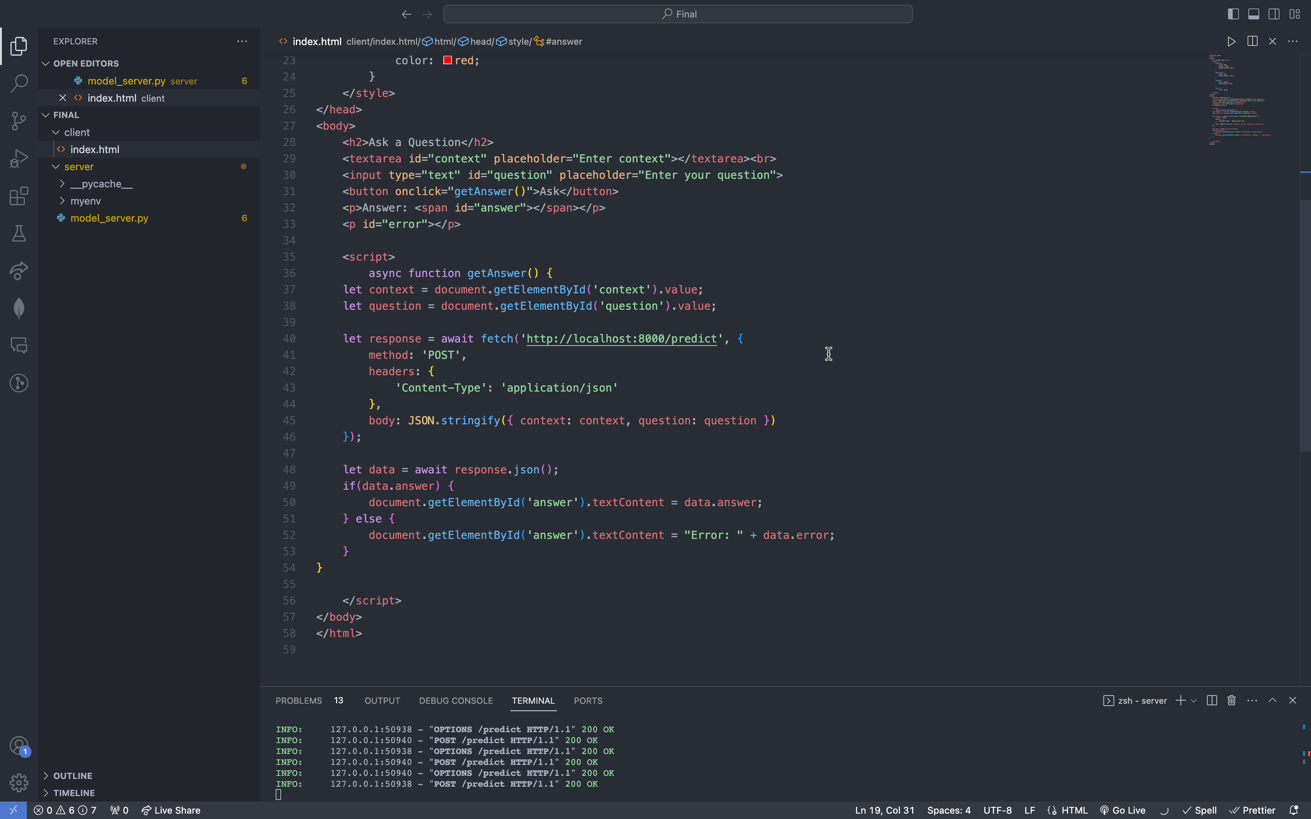
**Response**

The API will return a JSON response with the following structure:

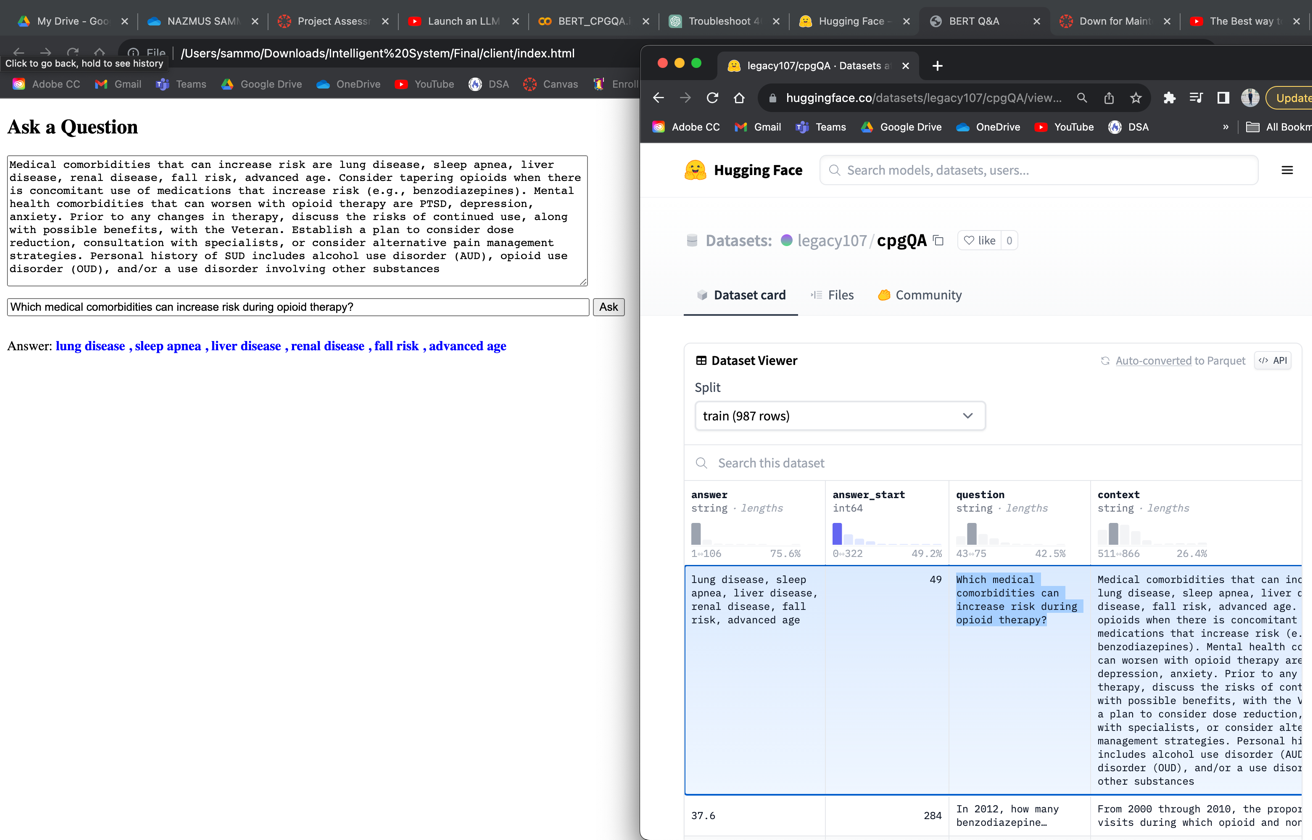
{ "answer": "Extracted answer from the context..." }

* **answer** (string): The extracted answer from the provided context.

**And the Code for the frontend is:**



With everything done the output of the website is:



That’s everything for the fortnightly report. Thanks.