# Fine-tune LLM for Code Generation

Afnan Aghai
RPTU Kaiserslautern, Department of Computer Science

***Note:*** *This report contains a project documentation and reflection on the portfolio task submitted for the lecture Engineering with Generative AI in WiSe 2023-24. This report is an original work and will be scrutinised for plagiarism and potential LLM use.*

## 1 Documentation

### 1.1 Dataset Selection

For the portfolio task of Engineering with Generative AI, I chose the dataset from the HuggingFace library by the name of **code_instructions_120k_alpaca** by a huggingface user **iamtarun** [1]. Dataset contains four columns by the names of **instruction**, **input**, **output**, **prompt** and has **121,959** rows. During the dataset selection process, I was particularly looking for a dataset that had instructions related to Python code generation and prompts that would contain Python code snippets along with a brief explanation of the code itself. Secondly, I tried to find a dataset that was already clean including clear and concise instructions and clean and efficient Python code. Thirdly, I wanted the dataset to have code snippets on multiple and diverse topics. This diversity would come in handy when training the model. After going through a range of publicly available datasets, I shortlisted 3 datasets, **code_instructions_120k_alpaca**, **evol-codealpaca-v1** by **theblackcat102** [2] and **starcoderdata** [3]. I didn't choose the Starcoder dataset because its size was not feasible and it also had programming languages other than Python. And I chose code_instructions_120k_alpaca dataset instead of evol-codealpaca-v1 because evol-codealpaca-v1 had prompts that didn't have any code snippets but just explanations.

### 1.2 Model Selection

For the model selection part, I reviewed the models which were taught in class. I looked through the documentation and ways of fine-tuning available on the internet for **BERT** and **Llama-2**. I chose a version of Llama-2-7B which was used for chat generation. It was on HuggingFace by the name of **Llama-2-7b-chat-hf** by **NousResearch** [4]. The reason for choosing Llama-2-7B was because firstly, it was taught to us in more detail through exercises so I generally had a better idea about Llama-2. Secondly, other models like Llama 2 70B performed better when asked to generate Python code snippets [5] [6]. I personally tested with the prompt "add 2 numbers" and seldomly models above Llama 2 7B would generate a sample code snippet but I didn't notice this in Llama-2 7B. In this way, this fitted the requirement of the portfolio task.

## 1.3 Design Overview

After selecting the dataset and model for the project, a suitable strategy for fine-tuning the model, splitting the datasets, generation of prompts and evaluation metrics was to be chosen. After reading through the lecture slides and exercises, the Following strategies were opted for.

## 1.4 Dataset Splitting

Dataset splitting is considered an essential part of Machine Learning and Artificial Intelligence. There are many different techniques and ratios which split the datasets into training, testing, and validation. The chosen dataset for the portfolio task had a row count of 120k which would consume a lot of time in training the model. Due to this reason, the dataset was sliced and 600 rows were extracted for the training dataset. Selected rows were removed from the dataframe and the dataframe itself was shuffled to promote randomness, and then 75 rows were extracted for the testing dataset. Choosing a number for the training dataset was crucial because the model's training would be utilizing 1/4 of it for calculating the training loss.Secondly, in the later stages of the project, dataset synthesizing was required which would give a synthetic dataset 3 times bigger than the original training dataset. And thirdly, a limited number of API requests were provided to us for synthesizing the dataset. Keeping all these reasons in mind, the training dataset's number was to be chosen carefully to not be very large which would be problematic to synthesize, and not small enough which would result in overfitting or bad training. With this in mind, a training dataset which is sufficient enough quantitatively and qualitatively to fine-tune a base LLM model was acquired. Moreover, testing the dataset helped in gauging the model's performance after training. As the project progressed to further models, I increased the size of the datasets. My strategy was to increase the dataset on each training period so as to see if increasing the dataset affects the model training positively or negatively or if the model trained on the highest number of rows would yield the greatest accuracy according to the evaluation metric. because of these reasons, I got 800 training and 100 testing dataset rows for model C and 1200 training and 200 for testing dataset rows for model D.

## 1.5 Fine Tuning the Model

For the portfolio exam, I bought **100 compute units** on Google Collab. And decided to go for the **Supervised Fine Tuning approach** since this approach closely aligned with the requirements of the exam. This technique allowed me to customize a pre-trained LLM and fine-tune it specific to my needs which means no training from scratch. This would help in saving time and the compute units on Google Collab. For using this technique, some variables were set in the beginning with the help of **BitsnBytes** including learning rate, weight decay, and epoch size which was set to 1 to avoid overfitting. Next, the SFT trainer was initialized with the OriginalModel and training and testing datasets. This finetunes the original model and gives us Model B. After getting our Model B, I used the same values for the SFT trainer for training Model B,C and D except for the training and testing datasets.

## 1.6 Prompt Innovation

The main idea for me was generating a prompt that could test the model by requesting to do something that would generally not require a code but still model would give a code for it. I tried different prompts some of them are as follows:

1. **"Extract meaningful insights from the sentence "my name is Afnan.""** which would give a Python code using different libraries for giving insights into it.
2. **Do mathematical operation between 2 numbers** which would also give a Python code.
3. **Create a rectangle using 1's and 0's** Although this is a bit specific still it doesn't explicitly ask for a code snippet.

Apart from this, clear, concise, and straight-to-the-point prompts were needed during the data synthesizing process. It is because I wanted a dataset that had answers that would not contain unnecessary details or text. All these would clutter the dataset and would adversely affect the training process. So, I wrote prompts for generating dataset from the API which are as follows:

1. **"Following is a topic related to Python: " + topics[last_index] + " generate 30 coding questions only on this topic which are unique and often asked by people on the internet. These questions can be answered with clean and concise Python codes."**
   This would give me answers to the topics related to Python which I had stored in an array by the name of topics. Details about data synthesizing are in the **section 1.2.5**.
2. **"Given the question:" + synthesized_data_final.loc[i,'Question'] + "Please provide a Python code snippet or function along with an explanation similar to responses on Stack Overflow. Ensure your response is within 1950 characters. Don't write any unnecessary text."**
   This prompt was used when I wanted to generate answers of the questions which I got earlier. Details about data synthesizing are in the **section 1.2.5**.

These prompts generated answers which were short and to the point.

## 1.7 Synthesizing Dataset For Model C

Synthesizing the dataset was to some extent similar to exercise 3 in which we generated data using the provided API Key. The only differences were the amount of data which was generated and using innovative prompts for the generation. My initial training dataset number was 600 meaning that my synthesized dataset would have a number of 1800. So I found a list of Python topics on which coding questions could be generated. This list had 60 topics. My idea was to generate 30 questions for each topic which would yield a number of 1800 which was required. Now, to efficiently use the API Key, I generated 30 questions, using the prompt mentioned in **Section 1.2.4**, in each API request which constitutes 60 requests. I stored these questions in a CSV file and then I made 1800 API requests, each for generating an answer, using the prompt I mentioned in **Section 1.2.4**, to the question that was generated earlier. I stored the responses along with the questions in a CSV file. Later on, the synthesized dataset was split for training and testing and transformed through pandas so that it could be fed to the model for training.

## 1.8 Evaluation Metric Selection

There was a choice between evaluation metrics used for evaluating LLM's performance. This included Perplexity, BLEU, ROUGE etc.I tried implementing Perplexity but it gave an unusually high value. I believe it was because Perplexity generally is used for measuring the prediction power of the model as to how accurately can the model predict the text. Perplexity score tells how much the model is surprised when the original data is shown to it. But here we were dealing with code snippets, which means that if the original response to an instruction was a code snippet containing "a+b" and the model would say "c+d", the perplexity measure would go up since it would assume that both these answers are different. Although both of these are logically the same. On the other hand, BLEU was a metric used to measure the quality of the text which is generated by the model. It was closely related to the idea of evaluation in this scenario which was to check the quality of the code being generated by the model and the quality of the overall response.

## 1.9 Functionality & Code

Notebook starts off with traditional imports and pip installs which are to be used later. The first task which is addressed in the notebook is loading the dataset from huggingface. It is loaded and then transformed in a format suitable for model training using Pandas. The dataset is split and training and testing datasets are stored as CSV files in google drive. Then comes the loading part where the original model is loaded from huggingface. This model is then Finetuned using the training dataset we got earlier using the SFT Approach discussed in previous sections and we then get Model B's adapters. Adapters are stored on google drive and in the next step, these adapters are merged with the base model to get **Model B**. Model B is then uploaded on huggingface so that it can be easily unloaded in the notebook in case of a system or notebook crash.

Code then deals with the data synthesizing part where the API key is stored in a variable. Along with this, a list of 60 Python topics is also stored in the notebook through which synthesized dataset for model C will be generated. 2 Functions containing prompts for both questions and answers respectively come next. These functions are called in a for loop and the responses are stored in a CSV file directly to avoid any hassle if notebook crash occurs.

CSV file is then loaded and like previously, dataset is split and transformed for model training. The whole process of training and then merging the adapters is repeated and model C is then acquired. Model C is then uploaded on huggingface for the same reasons.

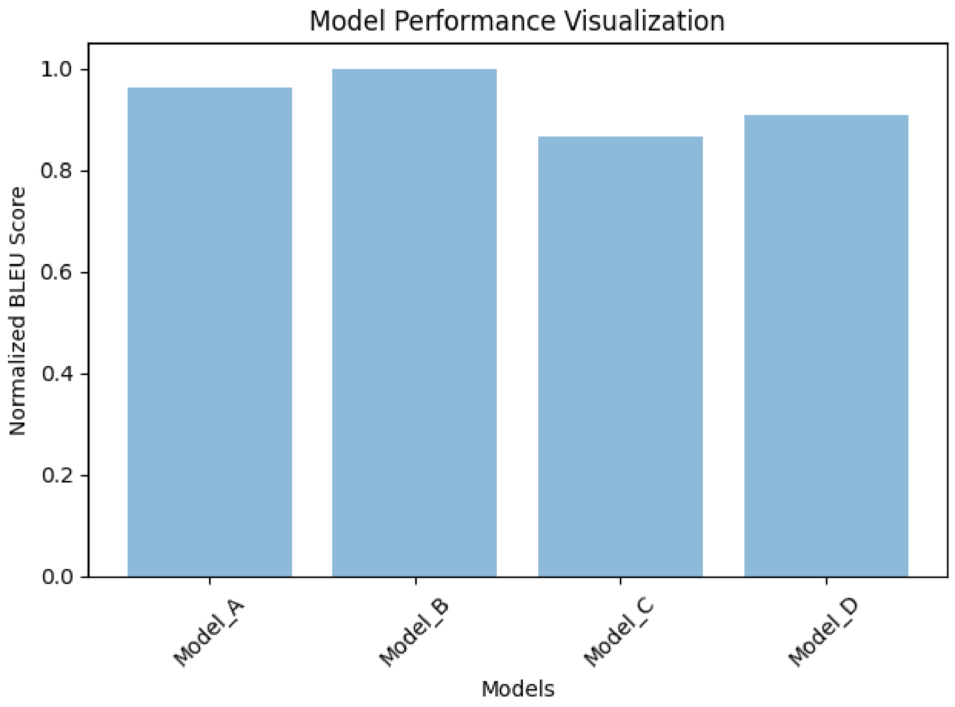
Next, there are different functions like unloading the model from huggingface or removing any extra content from the answers to the questions in the CSV file. Then comes the merging part where synthesized dataset and initial training dataset are merged for training model D. synthesized dataset is split in training and testing and then merged with the initial training dataset and stored in CSV. Process which was done for model C and model B is also repeated for model D.

After this, the notebook has a section for evaluation where 2 metrics (one partial) have been implemented. 35 rows from the original dataset(not part of the initial training dataset) are extracted and stored separately. Afterward, each model is used for generating responses on the

35 instructions that were extracted earlier. These responses are stored in separate CSV's. Only model A's responses were used for Perplexity evaluation but due to wrong results and other reasons explained in previous sections, BLEU was used for evaluations. After calculating BLEU scores, these scores are plotted in a bar graph to provide a clear visualization of which model performs better.

### 1.10 Evaluation & Analysis

Looking at the Graph below, it is noticed that out of the 4 models, Model B performed well. This was the model trained on the initial training dataset while models C and D were trained on synthesized or a mixture of synthesized and original datasets. By taking all these points into consideration, we can deduce that the model may perform better if it is trained on a human-evaluated or generated dataset rather than training on a dataset which is generated by another LLM or AI model.



## 2 Reflection

**What was the most interesting thing that you learned while working on the portfolio? What aspects did you find interesting or surprising?**

For me, the most interesting part was the fine-tuning of the model. I never personally worked with LLM's except for exercise sessions which were held in class. So doing it personally for the first time was very intriguing to me. It fascinated me that a chat model could become something else if the right dataset is fed to it. This was a small dataset that did the job for me but MNC's datasets are huge because of this they are investing tons of money in this. Before this project, I believed that it was unnecessary but now I understand. Not only this, But the synthesizing dataset part was also very fascinating concept. In the future, these models will become self-sufficient to provide datasets close enough to human-generated datasets for other model training.

**Which part of the portfolio are you (most) proud of? Why? What were the challenges you faced and how did you overcome them?**

Synthesizing the dataset was the part which was really challenging for me. The reason was only a limited number of API requests could be made, so I had to use them carefully. To tackle this, I generated more data in 1 request like I generated 30 questions in 1 request. This kept the requests count lower and I was able to generate the dataset safely. Secondly, there was a chance that the request would send the same response. For e.g if i asked the model to generate coding question regarding functions in python, it would generate a question asking for a function xyz which is adding numbers. Now if i asked for a question on the same topic, there is a high chance that model API's response would have the same question. This is because it is not keeping track of the conversation I am having with the model. So in order to solve this dilemma, I generated all the questions regarding a particular topic in one go so that it all questions would be unique. What adjustments to your design and implementation were necessary during the implementation phase? What would you change or do differently if you had to do the portfolio task a second time? What would be potential areas for future improvement?

**Include a brief section on ethical considerations when using these models on code generation tasks.**

The datasets and models that I have used in this project are all publicly available. I have put references for each of them in the document. Moreover, If there is a code snippet that I have taken from somewhere on the internet, I have added a reference for it too. Furthermore, If there is a question as to the usage of these models in daily routine for code generation or something related to academics, I believe that models like ChatGPT could be utilised to improve the minor bugs in the code which is written by yourself rather than just asking to generate the whole code. If used correctly and ethically, these models could be great for pair programming for e.g: GitHub CoPilot. Lastly, creating a sole dependency on these models may result in unemployment in near future since these models are able to work tremendously well in some of the fields. These uses should be regulated in a way which reduces the risk of unemployment.

**From the lecture/course including guest lectures, what topic excited you the most? Why? What would you like to learn more about and why?**

Guest lectures from the AWS people were the ones that had really interesting topics. One of the lectures showed a technique of how symmetrical (maybe wrong word) adapters could be used in fine-tuning a model with fewer computations by freezing the model weights and using adapters. I would like to delve deeper into the topics like RAG.

**Based on the content of the lecture taught during the semester and the task you have carried out in this portfolio, describe a project that you would like to do using generative AI and LLMs**.

If I get the chance, I would like to try to do Task 1 of the project. Both topics which have been covered in task 1 and 2 are interesting to me. Not only this, field of generative AI itself is booming so learning as much about this field would prove to be beneficial for me. So for me, I would like to do task 1 of the portfolio exam.

**What would be a good multiple-choice question for a test to see if a student has really understood the content? Design 3 Multiple Choice Questions (with at least 3 answer choices)**

1. We need to evaluate a chat LLM model which generates humanly text and can chat with the user. Evaluation should be done of how close the result of the model is as compared to actual response. Which evaluation should you choose?
   - Perplexity
   - BLEU
   - ROUGE
2. When integrating LLM model in a web application which is hosted on a remote server, what are the possible issues which needs to be addressed regarding this integration(there can be more than one option):
   - Keeping the response time to the minimal
   - Management of resources related to the response handling of the LLM.
   - Ensuring that integration won't affect compatibility with the browsers.
3. In the process of Reinforcement Learning(RL), what are the weaknesses of Vanilla Policy Gradient(VPG) which are covered by Proximal Policy Optimization(PPO):
   - VPG requires a greater amount of data to achieve optimal performance.
   - VPG uses an adaptive scheme to tweak the learning rate.
   - It is the other way around, VPG is better than PPO.

## References

[1] iamtarun. iamtarun/code_instructions_120k_alpaca.

[2] theblackcat102. theblackcat102/evol-codealpaca-v1.

[3] bigcode. bigcode/starcoderdata.

[4] NousResearch. Nousresearch/llama-2-7b-chat-hf.

[5] Eduardo Muñoz. Fine-tuning a llama-2 7b model for python code generation.

[6] misc. Llama 2 vs bert.

# Informed Consent of Participation

Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

You are invited to participate in the field study **LLM Education** initiated and conducted by Applied Machine Learning group. The research is supervised by **Sebastian J Vollmer**. Please note:

- Your participation is entirely voluntary and can be withdrawn at any time.
- The field study will last approximately 6 weeks.
- We will record the documentation submitted by the student at the end of the portfolio exam.
- All records and data will be subject to standard data use policies.

If you have any questions or complaints about the whole informed consent process please contact Sebastian J Vollmer (E-Mail: sebastian.vollmer@dfki.de).

## Purpose and Goal of this Research

Ability to create multiple choice questions of LLMs. Can we prompt an LLM to create good multiple choice questions? Your participation will help us achieve this goal. The results of this research may be presented at scientific or professional meetings or published in scientific proceedings and journals.

## Participation and Compensation

Your participation in this study is completely voluntary and is unpaid.

## Procedure

After confirming the informed consent the procedure is as follows:

1. Student submits the exam with deliverables
2. Student also creates questions of their choice based on instructions of good multiple choice questions.
3. We compare LLM's output to MCQs created by students.

The complete procedure of this field study will last approximately 6 weeks.

## Risks and Benefits

There are no risks associated with this field study. We hope that the information obtained from your participation may help to bring forward the research in this field. The confirmation of participation in this study can be obtained directly from the researchers.

## Data Protection and Confidentiality

We are planning to publish our results from this and other sessions in scientific articles or other media. These publications will neither include your name nor cannot be associated with your identity. Any demographic information will be published anonymized and in aggregated form. Contact details (such as e-mails) can be used to track potential infection chains or to send you further details about the research. Your contact details will not be passed on to other third parties. Any data or information obtained in this field study will be treated confidentially, will be saved encrypted, and cannot be viewed by anyone outside this research project unless we have you sign a separate permission form allowing us to use them. All data you provide in this field study will be subject of the General Data Protection Regulation (GDPR) of the European Union (EU) and treated in compliance with the GDPR. Faculty and administrators from the campus will not have access to raw data or transcripts. This precaution will prevent your individual comments from having any negative repercussions. During the study, we log experimental data, and take notes during the field study. Raw data and material will be retained securely and compliance with the GDPR, for no longer than necessary or if you contact the researchers to destroy or delete them immediately. As with any publication or online-related activity, the risk of a breach of confidentiality or anonymity is always possible. According to the GDPR, the researchers will inform the participant if a breach of confidential data was detected.

## Identification of Investigators

If you have any questions or concerns about the research, please feel free to contact:
Sebastian J Vollmer
Principal Investigator Trippstadter Str. 122
67663 Kaiserslautern, Germany sebastian.vollmer@dfki.de

☑ I understand the explanation provided to me. I understand and will follow the hygiene rules of the institution. I understand that this declaration of consent is revocable at any time. I have been given a copy of this form. I have had all my questions answered to my satisfaction, and I voluntarily agree to participate in this field study.

☑ I agree that the researchers will and take notes during the field study. I understand that all data will be treated confidentially and in compliance with the GDPR. I understand that the material will be anonymized and cannot be associated with my name. I understand that full anonymity cannot be guaranteed and a breach of confidentiality is always possible. From the consent of publication, I cannot derive any rights (such as any explicit acknowledgment, financial benefit, or co-authorship). I understand that the material can be published worldwide and may be the subject of a press release linked to social media or other promotional activities. Before publication, I can revoke my consent at any time. Once the material has been committed to publication it will not be possible to revoke the consent.

**Signature**: Afnan Aghai **Place, Date**: Kaiserslautern, 29.04.2024