

# Configuration Manual

MSc Research Project  
MSc of Artificial Intelligence

Muhammad Hassan Shakeel

Student ID: 23215992

School of Computing  
National College of Ireland

Supervisor: Dr. Anh Duong Trinh

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	Muhammad Hassan Shakeel
<b>Student ID:</b>	23215992
<b>Programme:</b>	MSc of Artificial Intelligence
<b>Year:</b>	2024
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Dr. Anh Duong Trinh
<b>Submission Due Date:</b>	18/12/2024
<b>Project Title:</b>	Configuration Manual
<b>Word Count:</b>	484
<b>Page Count:</b>	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

<b>Signature:</b>	Muhammad Hassan Shakeel
<b>Date:</b>	24th January 2025

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

Attach a completed copy of this sheet to each project (including multiple copies).	<input checked="" type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input checked="" type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input checked="" type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Muhammad Hassan Shakeel  
23215992

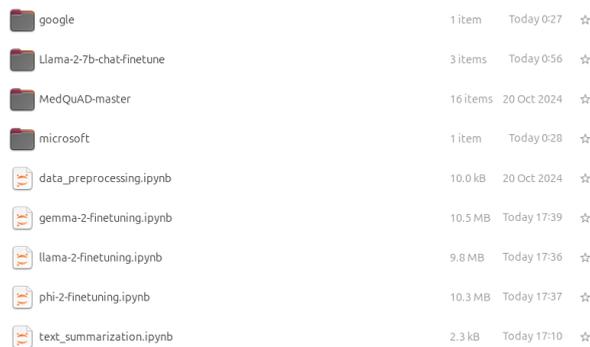
## 1 System Requirements

- Use Google Colab pro for fine-tuning and evaluation of LLM models
- Select L4 GPU with "High RAM" toggle button enabled
- The selected environment should have 53 GB of RAM, 22.5 GB of GPU RAM and 235.7 GB of total disk space.

## 2 Data Acquisition

Although data is available in "MedQuAD-master" directory inside the zipped directory. But dataset is publicly available to download as well at <https://github.com/abachaa/MedQuAD> (Accessed at: 17/12/2024) (Abacha; 2024).

## 3 Project Structure



google	1 item	Today 0:27	☆
Llama-2-7b-chat-finetune	3 items	Today 0:56	☆
MedQuAD-master	16 items	20 Oct 2024	☆
microsoft	1 item	Today 0:28	☆
data_preprocessing.ipynb	10.0 kB	20 Oct 2024	☆
gemma-2-finetuning.ipynb	10.5 MB	Today 17:39	☆
llama-2-finetuning.ipynb	9.8 MB	Today 17:36	☆
phi-2-finetuning.ipynb	10.3 MB	Today 17:37	☆
text_summarization.ipynb	2.3 kB	Today 17:10	☆

Figure 1: Project Structure

The project structure shown in Fig. 1 is explained below:

- Directories "google", "Llama-2-7b-chat-finetune" and "microsoft" contains fine-tuned model parameter configurations that we can load in our code files and use later.
- "MedQuAD-master" directory contains the raw dataset we downloaded from GitHub link shared earlier.

- The dataset is in xml format, therefore the file data-preprocessing.ipynb converts data into CSV format and it also filters out data rows that contain empty string in the answer column.
- Afterwards, text\_summarization.ipynb file is executed to generate extractive summary of the answers provided in the dataset.
- By this point data preprocessing is done, so gemma-2-finetuning.ipynb, llama-2-finetuning.ipynb and phi-2-finetuning.ipynb files can be run in any order as well as it can be run in parallel as well.

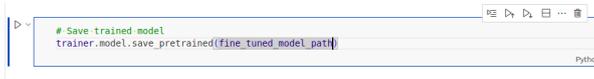
## 4 Code Execution Guidelines

### 4.1 Preprocessing guidelines

- Make sure to set current directory as root directory of the project and avoid changing any directory/file names.
- Execute data\_preprocessing.ipynb file cell-by-cell. When all the cells are executed in sequential order, a new file MedQuAD.csv will be saved in root directory of the project.
- Execute text\_summarization.ipynb file in sequential order to generate summary of the answers. Summaries will be appended in the CSV file that was generated by data\_preprocessing.ipynb in the last step.

### 4.2 Fine-tuning guidelines

- For fine-tuning of the LLM model, execute each of the three LLM model ipynb files until the cell shown in fig. 2. Execute the cell shown in the fig. 2 as well if you want to save the fine-tuned model configurations.
- For evaluation of LLM models, continue execution until the cell shown in fig. 3. Cell shown in the figure calculates BLEU score.



```

# Save trained model
trainer.model.save_pretrained(fine_tuned_model_path)

```

Figure 2: Cell - To save the fine-tuned model

```

!pip install nltk==3.8.1 # Install nltk if you haven't already

import nltk
from nltk.translate.bleu_score import sentence_bleu

# Download necessary data for nltk (if not already downloaded)
nltk.download('punkt')

# Calculate BLEU scores for each row in test_dataset
bleu_scores = []
for index, row in test_dataset.iterrows():
    reference = row['test_prompt'] # Ground truth
    candidate = row['generated_response'] # Model output

    # Tokenize reference and candidate
    reference_tokens = nltk.word_tokenize(reference)
    candidate_tokens = nltk.word_tokenize(candidate)

    # Calculate BLEU score (using smoothing function)
    score = sentence_bleu([reference_tokens], candidate_tokens, smoothing_function=nltk.translate.bleu_score.smoothing_function)
    bleu_scores.append(score)

# Print average BLEU score
avg_bleu = sum(bleu_scores) / len(bleu_scores)
print(f"Average BLEU Score: {avg_bleu}")

```

Figure 3: Cell - For evaluation of the LLM

### 4.3 Load Fine-tuned LLM

- If you have you not executed any cell of the LLM fine-tuning code files. Make sure to execute until and including the cell shown in fig. 4.
- Then execute the cell shown in the fig. 5 to load fine-tuned LLM.
- To evaluate the loaded model, execute three cells shown in the fig 6. Also, execute the cell shown in the fig. 3

```

# The model that you want to train from the Hugging Face hub
model_name = "google/gemma-2-2b"

# The instruction dataset to use
dataset_name = "mlabonne/guanaco-llama2-1k"

# Fine-tuned model name
new_model = "google/gemma-2-2b-finetuned"

#####
# QLoRA parameters
#####

# LoRA attention dimension
lora_r = 64

```

Figure 4: Prerequisite of loading saved model

```

# fine_tuned_model_path = './content/drive/MyDrive/Practicum/'

# Reload model in FP16 and merge it with LoRA weights
base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    low_cpu_mem_usage=True,
    return_dict=True,
    torch_dtype=torch.bfloat16,
    device_map=device_map,
)

model = PeftModel.from_pretrained(base_model, fine_tuned_model_path)
model = model.merge_and_unload()

# Reload tokenizer to save it
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"

```

Figure 5: Load Fine-tuned LLM

```
import locale
def getpreferredencoding(do_setlocale = True):
    """Return the preferred encoding, or 'UTF-8' if do_setlocale is True and
    locale.getpreferredencoding fails. """
    locale.getpreferredencoding = getpreferredencoding

import locale
locale.getpreferredencoding = lambda: "UTF-8"

!pip install rouge_score==0.1.2 # Install rouge_score
from rouge_score import rouge_scorer

# Initialize the ROUGE scorer
scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)

# Calculate ROUGE scores for each row in test_dataset
rouge_scores = []
for index, row in test_dataset.iterrows():
    scores = scorer.score(row['test_prompt'], row['generated_response'])
```

Figure 6: Evaluation steps of Fine-tuned model

For fine-tuning and loading fine-tuned LLM, same steps will be applied for all three LLM files.

## References

Abacha, A. B. (2024). Medquad: A medical question answering dataset, <https://github.com/abachaa/MedQuAD>. A comprehensive dataset for medical question answering.