

10 Academy Cohort B

Weekly Challenge: Week 6

**LLM Finetuning: Enabling Quality Embedding and Text Generation for Amharic, Swahili, and Yoruba Languages**

# Overview

Business Need

AfroTech Solutions is an innovative African company that focuses on using AI to improve customer support and engagement. Our goal is to help African businesses by using new technology in AI. Our latest project is an AI-powered customer support system designed for the African market. By using advanced AI, this system aims to provide smooth, multilingual support across different platforms.

This project aims to make our customer support services better by using **AI to generate text in Amharic, Swahili, and Yoruba**. We plan to create systems that can generate quick and relevant responses in these languages based on customer questions and past interactions.

For this project to be successful, our customer support must be both efficient and meaningful to our diverse clients. To achieve this, the technology needs to have strong capabilities in text embedding and generation for Amharic, Swahili, and Yoruba. We will collect large datasets for each language and fine-tune suitable open-source LLM models based on the datasets collected in the previous week. For Amharic, we may use models like Nous Hermes Mistral 8 7B or amharic language finetuned version of LLama 2 ([Samuael/llama-2-7b-tebot-amharic](https://huggingface.co/Samuael/llama-2-7b-tebot-amharic/tree/main) or [iocuydi/llama-2-amharic-3784m](https://medium.com/@garrilogistics/llama-2-amharic-llms-for-low-resource-languages-d6fb0ba332f4)) - and finetune it further to deliver the business objective. We will also select and fine-tune equivalent models for Swahili and Yoruba based on the collected data and the chosen language in the previous week to meet our business goals effectively.

# 

# Inspirations

The following works are our inspiration. We envision to collaborate with all stakeholders to create a robust quality LLM for the languages.

* [Llama2-Chinese/README\_EN.md at main · FlagAlpha/Llama2-Chinese (github.com)](https://github.com/FlagAlpha/Llama2-Chinese/blob/main/README_EN.md)

# Data

You will use the data you collected in the previous week.

# Expected Outcomes

## Skills:

* Experience working with Huggingface APIs and platform
* Fine-tuning and deploying LLMs
* Experience in using multiple GPUs for parallel training and inference
* Working with Deep Learning Frameworks
* Amharic, Swahili, and Yoruba text processing
* Proficiency in Python programming language
* Proficiency in Prompt Engineering

## Knowledge:

* Understanding Transformer Models and their components
* Understanding the building blocks of Instruction Based LLMs
* Understanding Chat Models such as ChatML chat template
* Natural Language Processing (NLP) Knowledge
* Machine Learning and AI Knowledge

# Team

Tutors:

* Yabebal
* Emitinan
* Rehmet

# Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

**Visualization** - quality of visualizations, understandability, skimmability, choice of visualization

**Quality of code** - reliability, maintainability, efficiency, commenting - in future this will be [CICD](https://en.wikipedia.org/wiki/CI/CD)/CML

**Innovative approach to analysis** -using latest algorithms, adding in research paper content and other innovative approaches

**Writing and presentation** - clarity of written outputs, clarity of slides, overall production value

**Most supportive in the community** - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

# Group Work Policy

This week, you are expected to complete the project with your assigned group. In the table below, your name is assigned to one of the groups we formed.

| Group Name (number) | Group Members | | Language | Machine Hostname |
| --- | --- | --- | --- | --- |
| Name | Email |
| Group 1 | Abdelrhman Yasir | [abdalrhmanalsh6@gmail.com](mailto:abdalrhmanalsh6@gmail.com) |  | [g1.10academy.org](http://g1.10academy.org) |
| Fatai Abdusalam | [abdusalamfatai60@gmail.com](mailto:abdusalamfatai60@gmail.com) |
| Getachew Abebe Agegnehu | [gechachin@gmail.com](mailto:gechachin@gmail.com) |
| Mistir Nigusse | [mistirnigusse0@gmail.com](mailto:mistirnigusse0@gmail.com) |
| Yohanes Teshome Kebede | [johnteshe13@gmail.com](mailto:johnteshe13@gmail.com) |
| Group 2 | Abraham Teka | [aberhamyirsaw@gmail.com](mailto:aberhamyirsaw@gmail.com) |  | [g2.10academy.org](http://g2.10academy.org) |
| Grace Nyutu | [graceynyutu@gmail.com](mailto:graceynyutu@gmail.com) |
| Melaku Alehegn | [melakualehegn34@gmail.com](mailto:melakualehegn34@gmail.com) |
| Henock Dessalegn | [henockgebriel@gmail.com](mailto:henockgebriel@gmail.com) |
| Group 3 | Abubeker Shamil | [abubekershamil123@gmail.com](mailto:abubekershamil123@gmail.com) |  | [g3.10academy.org](http://g3.10academy.org) |
| Michael George | [mikimare21@gmail.com](mailto:mikimare21@gmail.com) |
| Nyamusi Moraa | [nyamusiontita@gmail.com](mailto:nyamusiontita@gmail.com) |
| Eyerusalem Admassu | [eyerusad12@gmail.com](mailto:eyerusad12@gmail.com) |
| Group 4 | Temesgen Gebreabzgi | [temesgengebreab33@gmail.com](mailto:temesgengebreab33@gmail.com) |  | [g4.10academy.org](http://g4.10academy.org) |
| Addisu Alemu | [addisualemuj@gmail.com](mailto:addisualemuj@gmail.com) |
| Hillary Kipkemoi | [hillary6k@gmail.com](mailto:hillary6k@gmail.com) |
| Mahbubah Abdulhakim | [mahbubah.abdulhakim12@gmail.com](mailto:mahbubah.abdulhakim12@gmail.com) |
| Wandera Martin | [martinwandera@gmail.com](mailto:martinwandera@gmail.com) |
| Gilbert Tcheugoue Djissitchedjibril | [djibriltcheugoue@gmail.com](mailto:djibriltcheugoue@gmail.com) |
| Group 5 | Bethelhem Mebratu | [baerem123@gmail.com](mailto:baerem123@gmail.com) |  | [g5.10academy.org](http://g5.10academy.org) |
| Jabez Kassa | [jabezkassa2022@gmail.com](mailto:jabezkassa2022@gmail.com) |
| Samson Leul | [samsonleul@gmail.com](mailto:samsonleul@gmail.com) |
| Tewodros Cheru | [tewodroscheru9@gmail.com](mailto:tewodroscheru9@gmail.com) |
| Grace Ogundaini | [graceogundaini@gmail.com](mailto:graceogundaini@gmail.com) |
| Group 6 | Daisy Cherono | [jepchumbadaisy96@gmail.com](mailto:jepchumbadaisy96@gmail.com) |  | [g6.10academy.org](http://g6.10academy.org) |
| Dereje Hinsermu Senbatu | [derejehinsermu2@gmail.com](mailto:derejehinsermu2@gmail.com) |
| Joseph Njuguna | [josepnjuguuna@gmail.com](mailto:josepnjuguuna@gmail.com) |
| Selamawit Tibebu | [selamt4709@gmail.com](mailto:selamt4709@gmail.com) |
| Sheila Murugi | [tmurugi49@gmail.com](mailto:tmurugi49@gmail.com) |
| Group 7 | Hanna Mengistu | [hannasisay05@gmail.com](mailto:hannasisay05@gmail.com) |  | [g7.10academy.org](http://g6.10academy.org) |
| Mihret Kochito Wolde | [mihret.kochito16@gmail.com](mailto:mihret.kochito16@gmail.com) |
| Ermias Asmare | [it.ermias.asmare@gmail.com](mailto:it.ermias.asmare@gmail.com) |
| Dawit Mulie | [daveget2020@gmail.com](mailto:daveget2020@gmail.com) |
| Mikias Wondim Gebre Medhin | [mikiaswondim0962939288@gmail.com](mailto:mikiaswondim0962939288@gmail.com) |
| Dawit Wubale | [dawittegegnwork@gmail.com](mailto:dawittegegnwork@gmail.com) |
| Abraham Sahile | [abresh.agit@gmail.com](mailto:abresh.agit@gmail.com) |
| Biniyam Teshome | [se.biniyam.teshome@gmail.com](mailto:se.biniyam.teshome@gmail.com) |
| Blen Sleshi | [billu55sleshi@gmail.com](mailto:billu55sleshi@gmail.com) |
| Jolly Omere | [jollyomere@gmail.com](mailto:jollyomere@gmail.com) |
| Selam Yoseph | [selamk499@gmail.com](mailto:selamk499@gmail.com) |
| Koomi Toussaint Amoussouvi | [toussaint.amoussouvi@yahoo.com](mailto:toussaint.amoussouvi@yahoo.com) |

# Instructions

The rapidly evolving landscape of LLMs benefitted from the unprecedented scales of model size and training data producing models with strong capabilities, including reasoning and learning from experience at levels surpassing humans’.

However, due to the high imbalance in training data (text sources from the internet), English dominates in these models. Models are not as proficient in other languages, especially low-resource languages that are absent from the multilingual training corpora.

Collecting large-scale data for a low-resource language and retraining an LLM can be prohibitively expensive due to computational and data collection costs. A better approach is transfer learning; transferring an LLM’s capabilities from English to a non-English language through further pre-training and fine-tuning.

As part of this challenge, you are required to do the following tasks

* Understand the LLM landscape as of Jan 2024
* Understand the building blocks of
  + LLM base models (encoder only, encoder-decoder, decoder only)
  + GPU memory needs (15-20GB, 20-80GB, >80GB), Numbers of GPUs, and time to full pretraining and finetuning.
  + How to finetune
    - Instruct finetuning
    - Chat finetuning
* Understand the key components of LLM Training and Finetuning
  + Pre-training: self-supervised learning predicting the next word in a given context
  + Supervised fine tuning (SFT)
  + Parameter-efficient Tuning (PEFT)
  + Low-Rank Adaptation (LoRA)
* Overview of best contender open source LLM models and their variations
  + Mistral: 7B, 8x 7b,
  + Llama 2/3: 7b
  + Falcon: 7b
  + Stable AI 2: 1.6B
  + OpenLLama: 3B, 7B
* Explore huggingface documentation for inference and finetuning
  + Test hugging face embedding examples on your local machine tasks
  + Test hugging face small LLM models on your local machine
  + Test hugging face modules for
    - Data loading, preprocessing, batching, and tokenizing
    - Loading quantized models (BitsAndBytes)
    - Applying parameter-efficient finetuining (PEFT)
    - Applying LoRA
    - General techniques to reduce memory and finetune efficiently with a optimal tradeoff among, memory, speed, and accuracy
* Understand and Prepare the collected data for finetuning
  + Explore provided data as well as what you can find in the web
  + Prepare data to be ingested in your instruction-finetuning pipeline
  + Prepare evaluation datasets to benchmark your finetuned model with baseline OpenAI/Huggingface deployed models
* Select an open-source LLM model and finetune it

## Task 1: Literature Review & Hugging face ecosystem

In this task you are expected to review basic concepts and methods used to perform further pre-training and fine-tuning of a LLM:

* Get good understanding of the following concepts and techniques relevant to LLMs
  + Definitions of key terms and concepts
    - [hackerllama - The Llama Hitchiking Guide to Local LLMs (osanseviero.github.io)](https://osanseviero.github.io/hackerllama/blog/posts/hitchhiker_guide/)
  + Background knowledge on LLMs:
    - [Introduction to Large Language Models](https://www.lakera.ai/blog/large-language-models-guide)
    - [Understanding LLMs: A Comprehensive Overview from Training to Inference](https://arxiv.org/pdf/2401.02038.pdf)
  + Transformer architecture (encoder, decoder, self-attention)
    - [Transformers made easy: architecture and data flow | by Maâli Mnasri | Opla | Medium](https://medium.com/opla/transformers-made-easy-architecture-and-data-flow-f79f11961942)
    - [How Self Attention works in Transformer](https://medium.com/analytics-vidhya/how-self-attention-works-in-transformer-6c76a12396b0)
    - [Generative AI: The Science Behind Large Language Models - Simplified](https://www.linkedin.com/pulse/generative-ai-science-behind-large-language-models-aruna-pattam)
  + LLM Landscape
    - Architecture: three categories: Encoder-only , Encoderdecoder and Decoder-only
      * [LLM Boxing • Choose your Champion](https://llmboxing.com/leaderboard)
      * [Comparing the Best Open-Source Large Language Models | Shakudo](https://www.shakudo.io/blog/comparing-opensource-large-language-models)
      * [library (ollama.ai)](https://ollama.ai/library)
  + Embedding of the input data: Tokenization, Positional Embedding
    - [google/sentencepiece: Unsupervised text tokenizer for Neural Network-based text generation. (github.com)](https://github.com/google/sentencepiece)
    - [Test how tokenization works with OpenAI tokenizer](https://platform.openai.com/tokenizer)
    - [Master Positional Encoding: Part I | by Jonathan Kernes | Towards Data Science](https://towardsdatascience.com/master-positional-encoding-part-i-63c05d90a0c3)
    - [Understanding positional embeddings in transformer models (harrisonpim.com)](https://harrisonpim.com/blog/understanding-positional-embeddings-in-transformer-models)
  + Key concepts in Fine-tuning an LLM
    - [Fine-Tuning Embedding Model with PEFT and LoRA](https://medium.com/@kelvin.lu.au/fine-tuning-embedding-model-with-peft-and-lora-3b6f08987c24)
    - [Hosting A Text Embedding Model That is Better, Cheaper, and Faster Than OpenAI’s Solution](https://medium.com/@kelvin.lu.au/hosting-a-text-embedding-model-that-is-better-cheaper-and-faster-than-openais-solution-7675d8e7cab2)
* Get familiar with Huggingface ecosystem
  + Review huggingface documentation
    - [Hugging Face Hub documentation](https://huggingface.co/docs/hub/index)
    - [🤗 Transformers (huggingface.co)](https://huggingface.co/docs/transformers/index)
    - [Templates for Chat Models (huggingface.co)](https://huggingface.co/docs/transformers/main/chat_templating)
  + Try some examples on your local machine
    - [Building a PDF Knowledge Bot With Open-Source LLMs - A Step-by-Step Guide | Shakudo](https://www.shakudo.io/blog/build-pdf-bot-open-source-llms)
    - [An Introduction to Using Transformers and Hugging Face](https://www.datacamp.com/tutorial/an-introduction-to-using-transformers-and-hugging-face)
  + How instruction finetuning works
    - [huggingface/alignment-handbook: Robust recipes for to align language models with human and AI preferences (github.com)](https://github.com/huggingface/alignment-handbook)
* Amharic, Yoruba, and Swahili data collection, processing, and pipeline for LLM finetuning
  + [Llama-2-Amharic: LLMs for Low Resource Languages | by Garri Logistics | Dec, 2023 | Medium](https://medium.com/@garrilogistics/llama-2-amharic-llms-for-low-resource-languages-d6fb0ba332f4),
  + <https://huggingface.co/UBC-NLP/serengeti>
  + <https://huggingface.co/Mollel/Swahili_Gemma>
  + <https://huggingface.co/LeroyDyer/Mixtral_AI_SwahiliTron_7b>

## Task 2: Load an LLM and Use It for Inference

This task will see you setting up your work environment, load an open source pre-trained LLM and use it to generate output for a variety of scenarios (text generation, translation, question answering, summarization ..etc).

1. Set up a [Huggingface account](https://huggingface.co/join) (this is required for accessing some open source models e.g. LLaMA 2, and to upload your fine-tuned model later)
2. Note about [LLaMA](https://github.com/facebookresearch/llama?tab=readme-ov-file#access-on-hugging-face): to get access to the model you need to
   1. Fill this [Meta’s Form](https://ai.meta.com/resources/models-and-libraries/llama-downloads/), with the same email address you used to create your Hugging Face account.
   2. Visit the page of one of the LLaMA 2 available models, and accept Hugging Face’s licence terms and acceptable use policy.
3. Set up your environment [GPU enabled notebooks]
4. Make a choice of open source LLM to use.
   1. The choice of model (including the size of the model) depends on the use case and computational resource available.
      1. [Choosing the Right Open-Source LLM for Your Needs](https://www.datacamp.com/blog/top-open-source-llms#:~:text=90%25%20of%20cases.-,Choosing%20the%20Right%20Open%2DSource%20LLM%20for%20Your%20Needs,-The%20open%2Dsource)
      2. [Which Open Source LLM Best to FINE-TUNE ?](https://medium.com/@khalidreemy/which-open-source-llm-best-to-fine-tune-84333d3fea2f)
   2. Check the [comprehensive list](https://huggingface.co/blog/os-llms#:~:text=comprehensive%20table%20of%20some%20open%2Dsource/open%2Daccess%20models%20below) of open source/access LLMs on Hugging Face.
   3. [optional] You can use [HuggingChat](https://huggingface.co/chat/) (Hugging Face's open-source chat UI for LLMs) to check a couple of LLMs (eg: Mixtral-8x7B and Llama 2/3 model fine-tuned for dialogue)
5. Load an open source LLM from Hugging Face. The size of the model will depend on whether we use [quantization](https://huggingface.co/docs/transformers/quantization)
   1. Without model quantization:
      1. [How to download open source LLM models from hugging face and use it locally on your machine](https://medium.com/@anupamprataps_18643/how-to-download-open-source-llm-models-from-huggingface-and-use-it-locally-on-your-machine-1d21eb7cc47)
      2. [Running a Hugging Face Large Language Model](https://www.markhneedham.com/blog/2023/06/23/hugging-face-run-llm-model-locally-laptop/)
   2. With model quantization: [Model Quantization with 🤗 Hugging Face Transformers and Bitsandbytes Integration](https://medium.com/@rakeshrajpurohit/model-quantization-with-hugging-face-transformers-and-bitsandbytes-integration-b4c9983e8996)
6. Inference: use the loaded LLM to generate output. Make sure to test multiple inference scenarios (text generation, translation, question answering, summarization) and test the model’s ability to handle Amharic, Yoruba, and Swahili language
   1. [Hugging Face docs: Generation with LLMs](https://huggingface.co/docs/transformers/llm_tutorial#generation-with-llms)
   2. [optional] use a [pipeline for inference](https://huggingface.co/docs/transformers/pipeline_tutorial)

## Task 3: Data preprocessing and preparation

You may follow [Llama-2-Amharic: LLMs for Low Resource Languages | by Garri Logistics | Dec, 2023 | Medium](https://medium.com/@garrilogistics/llama-2-amharic-llms-for-low-resource-languages-d6fb0ba332f4) to understand the important steps of amharic data preparation for LLM finetuning.

Suggested tasks (be creative to do more) are:

* **For Amharic:**
  + \* ['ሐ', 'ሑ', 'ሒ', 'ሓ', 'ሔ', 'ሖ'] with ['ሀ', 'ሁ', 'ሂ', 'ሃ', 'ሄ', 'ህ', 'ሆ']
  + \* ['ኀ', 'ኁ', 'ኂ', 'ኃ', 'ኄ', 'ኅ', 'ኆ'] with ['ሀ', 'ሁ', 'ሂ', 'ሃ', 'ሄ', 'ህ', 'ሆ']
  + \* ['ሠ', 'ሡ', 'ሢ', 'ሣ', 'ሤ', 'ሦ', 'ሦ', 'ሧ'] with ['ሰ, 'ሱ', 'ሲ', 'ሳ', 'ሴ', 'ስ', 'ሶ', 'ሷ']
  + \* ['ዐ', 'ዑ', 'ዒ', 'ዓ', 'ዔ', 'ዕ', 'ዖ'] with ['አ', 'ኡ', 'ኢ', 'ኣ', 'ኤ', 'እ', 'ኦ']
  + \* ['ጸ', 'ጹ', 'ጺ', 'ጻ', 'ጼ', 'ጽ', 'ጾ'] with ['ፀ', 'ፁ', 'ፂ', 'ፃ', 'ፄ', 'ፅ', 'ፆ']
* **For Yoruba** 
  + Eliminate non-Yoruba text, HTML tags, special characters, and irrelevant content.
* **For Swahili**
  + Eliminate non-Swahili text, HTML tags, special characters, and irrelevant content.

## Task 4: Fine-Tuning the LLM

Steps needed to fine-tune the LLM. Steps from inputting the data to model deployment.

**For Amharic**

* Garri logistics model: [iocuydi/amharic-llama-llava (github.com)](https://github.com/iocuydi/amharic-llama-llava)
* [Chinese-LLaMA-Alpaca-2 v4.0](https://github.com/ymcui/Chinese-LLaMA-Alpaca-2) released long context LLMs (64K) and RLHF-tuned LLMs
* [facebookresearch/llama-recipes: Examples and recipes for Llama 2 model (github.com)](https://github.com/facebookresearch/llama-recipes)
* [alignment-handbook/scripts/README.md at main · huggingface/alignment-handbook (github.com)](https://github.com/huggingface/alignment-handbook/blob/main/scripts/README.md#fine-tuning-on-your-datasets)
* [How to Fine-tune an LLM Part 3: The HuggingFace Trainer | alpaca\_ft – Weights & Biases (wandb.ai)](https://wandb.ai/capecape/alpaca_ft/reports/How-to-Fine-tune-an-LLM-Part-3-The-HuggingFace-Trainer--Vmlldzo1OTEyNjMy) (check out part 1 & 2 as well)

as starter.

**For Swahili**

* [Mollel/Swahili\_Gemma](https://huggingface.co/Mollel/Swahili_Gemma)
* <https://huggingface.co/LeroyDyer/Mixtral_AI_SwahiliTron_7b>
* <https://huggingface.co/Mollel/Swahili-Alpaca-Llama-3-8b_16bit>

**For Yoruba**

* <https://huggingface.co/UBC-NLP/serengeti>

To use the Garri logistics model:

1. Accept Llama2 license on [huggingface](https://huggingface.co/meta-llama/Llama-2-7b-hf) and download it like this:
   1. *git lfs install*
   2. git clone <https://huggingface.co/meta-llama/Llama-2-7b-hf>
2. Download the amharic finetune from [huggingface](https://huggingface.co/iocuydi/llama-2-amharic-3784m) like this:
   1. *git lfs install*
   2. git clone <https://huggingface.co/iocuydi/llama-2-amharic-3784m>
3. Clone <https://github.com/iocuydi/amharic-llama-llava> repository
4. Then inside *inference/run\_inf.py*:
   1. comment the *import safety\_utils* line
   2. change the *MAIN\_PATH* to the path to folder you downloaded from step 1
   3. change the *peft\_model* to the path you cloned in the step 2
   4. go to your llama2 folder (from step 1) and replace all the tokenizer related files with the one you find from the 2nd step
   5. set *quanitzation=True* inside the main function before the *load\_model* function call
5. Finally run the *inference/run\_inf.py* file

You may choose one of the followng as your base model

* [Introducing Stable LM 2 1.6B — Stability AI](https://stability.ai/news/introducing-stable-lm-2)
* [Mistral 7B | Mistral AI | Open-weight models](https://mistral.ai/news/announcing-mistral-7b/) [2310.06825.pdf (arxiv.org)](https://arxiv.org/pdf/2310.06825.pdf)
* [imoneoi/openchat: OpenChat: Advancing Open-source Language Models with Imperfect Data (github.com)](https://github.com/imoneoi/openchat)
* [tiiuae/falcon-7b · Hugging Face](https://huggingface.co/tiiuae/falcon-7b)
* [Llama 2 - Meta AI](https://ai.meta.com/llama/)

## 

# Tutorials Schedule

In the following, the colour **purple** indicates morning sessions, and **blue** indicates afternoon sessions.

## Monday: The Mechanics of LLMs

Challenge walk through and Introduction to transformers.

* Introduction to Week Challenge (Yabebal)
* Overview of LLMs: Their transformer architecture and main techniques (Emtinan)

Key Performance Indicators:

* Understand the project
* Get a good understanding of how LLMs work

## Tuesday: Tokenization and Vocabulary Creation

More concepts

* Q&A session (Yabebal)
* Tokenization and Word embedding (Rehmet)

## Wednesday: LLMs Fine-tuning

Get a good understanding of data preparation and LLM finetuning .

* Different types of fine-tuning a pre-trained LLM (Emtinan)
* Components of a transformer Model (Emtinan)

## Thursday: Inference & LLMOps

More concepts

* Components of a transformer Model (Emtinan)
* Advanced use of Huggingface (Rehmet)

## Friday: Deployment

More concepts

* Data Preparation for Instruction Tuning (Rehmet)

# Deliverables

NOTE: Document should be a PDF stored in google drive or published blog link. **DO NOT SUBMIT A LINK as PDF!** If you want to submit pdf document, it should be the content of your report not a link.

## Interim Submission - Wednesday 8pm UTC

* Link to your code in GitHub
  + Repository where you will be using to complete the tasks in this week's challenge. A minimum requirement is that you have a well structured repository and some coding progress is made.
* A review report of your reading and understanding of Task 1 & 2 and any progress you made in other tasks.

#### Feedback

You may not receive detailed comments on your interim submission, but will receive a grade.

## Final Submission - Saturday 8pm UTC

* Link to your code in GitHub
  + Complete work for Finetuning LLMs with Amharic, Yoruba, and Swahili ***(Depends on which language you chose in the previous week)*** data
  + Complete work for Generating texts
* A blog post entry (which you can submit for example to Medium publishing) or a pdf report. .

#### Feedback

You will receive comments/feedback in addition to a grade.

# References

* [Complete Beginner’s Guide to Hugging Face LLM Tools](https://www.unite.ai/complete-beginners-guide-to-hugging-face-llm-tools/)
* [Making LLMs even more accessible with bitsandbytes, 4-bit quantization and QLoRA](https://huggingface.co/blog/4bit-transformers-bitsandbytes)
* [PEFT (Parameter-Efficient Fine-Tuning)](https://huggingface.co/docs/peft/index)
* [Finetuning Large Language Models (LLMs) BERT](https://www.analyticsvidhya.com/blog/2023/08/finetuning-large-language-models-llms/)

Infrastructure

* [Benchmarking Popular Opensource LLMs: Llama2, Falcon, and Mistral (truefoundry.com)](https://blog.truefoundry.com/benchmarking-llama2-falcon-and-mistral/)

# 