AI SINGAPORE

Post-Training/Evaluation

Technical Assessment

Deadline: **June 10, 2025**

Please note that submissions received after this date will not be considered.

🤗 | Overview | 🤗

This assessment is intended to explore your skills in post-training and evaluating large language models across the following three core components:

1. **Data Exploration and Generation**
2. **Instruction Fine-Tuning**
3. **Model Evaluation**

Introduction

The (fictitious) FinTech startup sea-fin.ai is looking to leverage LLMs as part of their market prediction workflow. Being able to identify the sentiment of any market related news, whether in English or in SEA languages, is key to their objective to corner the Southeast Asian market.

They have come to you for your help to develop a multilingual model that is tuned to identify the sentiment of any financial news. They want you to help to build an end-to-end pipeline that incorporates dataset processing, model training and model evaluation. This will help them to develop a fully functioning model that can determine the sentiment of the financial news.

Technical notes

This technical assessment can be fully completed in Kaggle. Resources you'll need minimally include **access to Kaggle (30 free weekly GPU hours)**—sufficient for completing all requirements without additional costs. **You are not supposed to purchase any additional computational resources.** You will also need a **HuggingFace account** to store the final model weights you have fine-tuned.

Given the resources available, we suggest using **Unsloth (github.com/unslothai/unsloth)** as your fine-tuning framework due to its focus on PEFT or parameter efficient fine-tuning (reducing the amount of compute necessary for training), but you are free to select whichever framework that you are most comfortable with.

To run a script on Kaggle, please refer to the Appendix.

Key deliverables

The following documents should be completed:

1. Jupyter notebook for any EDA performed
2. Python script for the data preparation pipeline
3. Python script for the synthetic data generation
4. Python script for the model training
5. Python script for the evaluation pipeline
6. Document containing your findings/thought processes for each decision as well as the responses to the discussion questions

While we expect a working model, we primarily value your approach towards experimentation.

Please **comment your thought process thoroughly** throughout your notebook as we are mainly interested in understanding how you tackle novel problems and innovate on existing research. This insight will give us a clearer picture of your unique strengths and areas of interest.

Please zip the completed files and email it to us before the deadline.

Task 1 | Data Exploration and Generation

Your manager at sea-fin.ai has come to you with a potential sentiment dataset in English. Given that they want a multilingual model, he is also suggesting that you check out the Vietnamese dataset that he has found. Additionally, he wants you to explore if it is possible to leverage LLMs to generate new sentiment datasets so that they can scale the amount of fine-tuning data that they have to work with. As with any LLM related task, you set off to explore the data at hand.

In this section, we will be curating and preparing a multilingual instruction dataset for **sentiment analysis** in English and in Vietnamese. You are free to use any tools, models, libraries and resources to assist you in building these datasets (but do keep in mind the limited compute available on Kaggle).

Deliverables for task 1

1. A training dataset of 6,500 instructions comprising 3,000 English instructions, 500 synthetically generated English instructions and 3,000 Vietnamese Instructions
2. An evaluation dataset of 500 instructions comprising 250 English instructions and 250 Vietnamese instructions
3. A synthetic data generation pipeline
4. EDA pipelines for English/Vietnamese
5. Answers to the discussion questions

Part 1.1 | Exploratory Data Analysis (English/Vietnamese)

You are provided with an English sentiment analysis dataset (**FinGPT/fingpt-sentiment-train**) and a Vietnamese sentiment analysis dataset (**uitnlp/vietnamese_students_feedback**).

Steps:

1. Perform an EDA on the provided English dataset, including standard descriptive statistics. Highlight any findings that could potentially impact model performance.
2. Using your findings from your EDA, clean the dataset (if necessary) and select **3,000** sentiments for training and **250** sentiments for evaluation.
3. Next, perform the EDA on the Vietnamese sentiment dataset. Highlight any findings that could potentially impact model performance.
4. Using your findings from your EDA, clean the dataset (if necessary) and select **3,000** sentiments for training and **250** sentiments for evaluation.
5. You should have a total of **6000 sentiments for training** and **500 sentiments for evaluation**.

Discussion questions:

1. Document down any differences that you have made in the Vietnamese EDA pipeline as compared to the English EDA pipeline and explain why.

Part 1.2 | Synthetic Data Generation (English only)

In this part, we will look to synthetically generate **at least 500 English instruction pairs**. You may choose to generate using either unsloth/Llama-3.2-1B or unsloth/Llama-3.2-1B-Instruct,

Generally, synthetic data generation comes in 3 parts: prompt creation, response generation and verification of the responses.

Prompt Creation: The prompts should be diverse, well structured and of varying complexity to simulate real-world inputs effectively.

Response Generation: Responses must be factually accurate, aligned with the intended tone, and formatted according to the user-defined output structure.

Response Validation: Verifications of the correctness and relevance of the responses generated are also a critical portion of synthetic data generation. Typically, these could include automated checks, human-in-the-loop validation, and alignment with predefined quality metrics to ensure the data is robust and reliable.

Steps:

1. Write a pipeline to create **500** diverse prompts with the model of your choice.
    a. This could be as direct as prompting the model directly to do so, or more involved with the inclusion of various seed texts.
2. Generate labels/responses for your prompts with the model of your choice.
    a. You can consider using various sampling strategies.
3. Implement a cleaning and verification step in the pipeline to ensure that the labels are correct.
4. Document down the various steps taken to create the prompts, responses and validation of the labels for the synthetic set. Provide the thought process and any findings that you have.

Discussion questions:

5. Why did you choose the model that you used to generate the synthetic data?
6. How would you ensure that the responses generated by the model follows in the specific format you desire?

Given the limited compute resources, we prioritize creative approaches and thoughtful validation strategies over quantity. You may focus on exploring effective methods rather than executing them at full scale. You may also consider limiting the sequence length for generation.

Task 2 | Supervised Fine-Tuning

With the fine tuning data at hand, you present your findings to your manager. He gives you the go ahead to start the training process. Unfortunately, there are constraints in the compute resources available. He tells you to train a 1B parameter LLM to see if it is suitable for the task at hand. He warns you that you might also need to employ some fine-tuning techniques to get it to train with the limited resources.

After completing basic data exploration and generation, you will proceed to fine-tune a small 1B model for an **Instruction-Tuning** task.

You can choose to fine-tune either unsloth/Llama-3.2-1B or unsloth/Llama-3.2-1B-Instruct for sentiment analysis. **Remember to document your reason for the choice.**

Deliverables for task 2

1. A pipeline to format the dataset into a format that is ready for model training
2. A pipeline to train the model
3. A final trained model
4. Answers to the discussion questions

Part 2.1 | Instruction Formatting

In this part, we will take the 3,500 English (including the 500 synthetic data you generated in the previous task) and 3,000 Vietnamese instructions that you have identified and process that data into a format ready for fine-tuning. This should be a reusable pipeline that can take in any dataset and output it in a format that is ready to be used for training.

Steps:

1. Build an ingestion pipeline to transform your data into a format ready for fine-tuning.
2. Print out **three** examples of their intermediate (in messages/conversations) and final form with chat template applied (showing all special tokens).

Part 2.2 | Supervised Fine Tuning

You will be using the instructions that you have formatted to train the model of your choice.

Steps:

1. Perform supervised fine-tuning of the model using the multilingual instruction pairs.
2. Upload and push your fine-tuned model to HuggingFace.

Discussion questions:

1. Why did you choose to train the model that you selected?

2. Unsloth uses PEFT to lower the computational resources needed to do fine-tuning. Compare the differences between PEFT and full fine-tuning.

Task 3 | Model Evaluation

The training process is done, you and your manager look at the loss function and it seems to have worked. However, he asks you to run some evaluations to verify that the model is doing what you have trained it for.

With your earlier selected evaluation dataset in Task 1, you will assess the performance of your fine-tuned model against the initial model that you have chosen, which serves as the baseline. Ideally, your fine-tuned model should demonstrate improved performance over the baseline, particularly in the languages you have fine-tuned it on.

Deliverables for task 3

1. An evaluation pipeline that takes in the evaluation dataset and runs inference on the model to determine the performance of the model.
2. Answers to the discussion questions

Steps:

1. Write a pipeline to
    a. Format your evaluation dataset into prompts
    b. Run inference on your fine-tuned model as well the original chosen model
    c. Parse the responses of the models
    d. Calculate metrics to compare the performance of both models
2. Document down the various steps taken to create the evaluation prompts and design choices for the evaluation pipeline

Discussion questions:

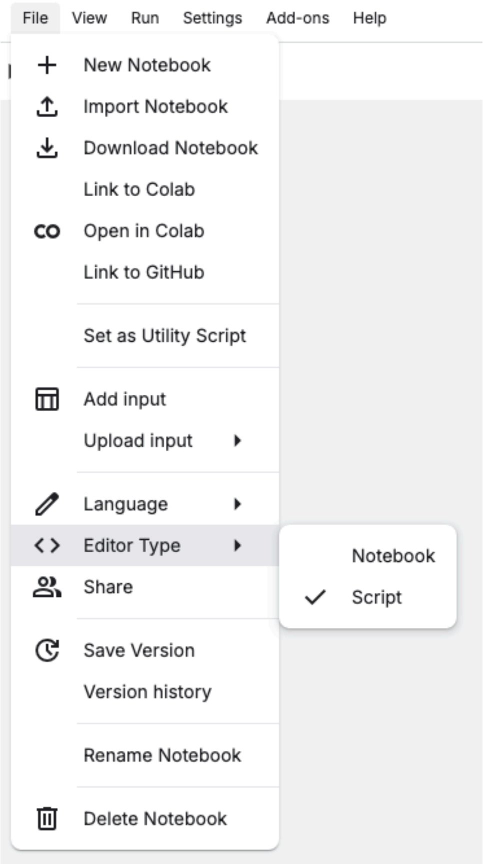
1. Report and discuss the differences in performances. Consider the issues with the instructions used in training, the actual fine-tuning process and running of evaluations.
2. Discuss any steps you took/can take to ensure that the various components of the pipeline (dataset choice, prompt choice, parser, metric) are providing a fair assessment of the models
3. Suggest improvements to the overall process (training and evaluation) and what you personally would like to explore next (assuming that time and compute is not a concern).

To streamline the evaluation process, we have predefined a test set using the datasets provided.

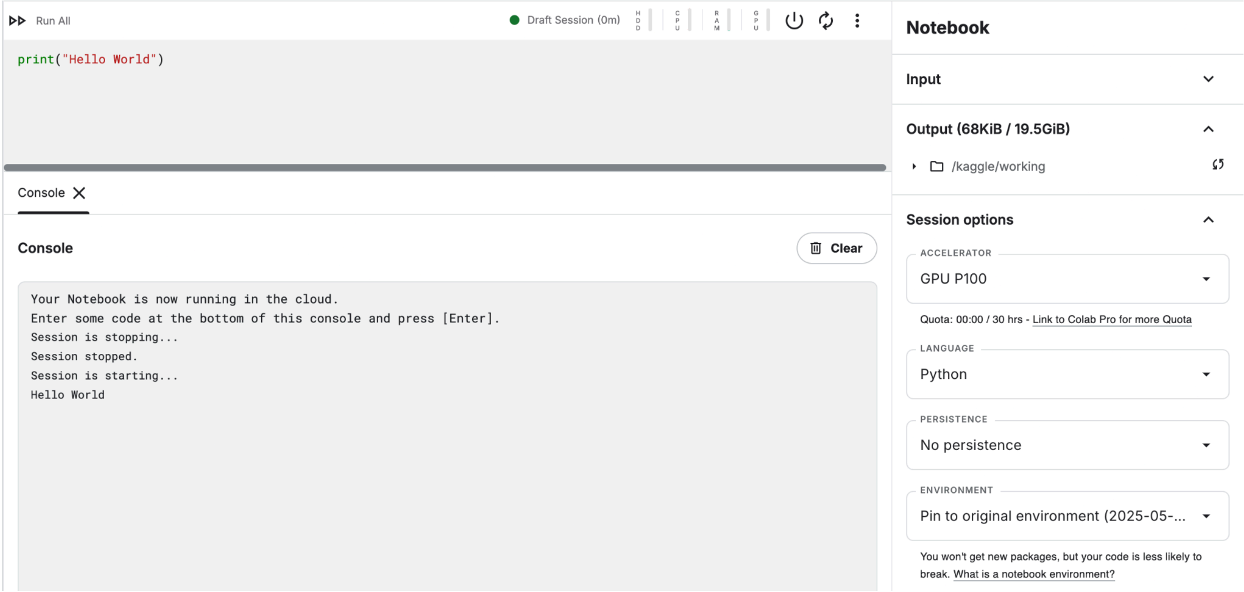
🤗 | Complete | 🤗

Appendix: How to run scripts in Kaggle

1. Select the editor type as *Script*



2. You can then click "Run All" to run the script



```
print("Hello World")
```

Console

```
Your Notebook is now running in the cloud.
Enter some code at the bottom of this console and press [Enter].
Session is stopping...
Session stopped.
Session is starting...
Hello World
```

Output (68KiB / 19.5GiB)

/kaggle/working

Session options

ACCELERATOR: GPU P100

Quota: 00:00 / 30 hrs - Link to Colab Pro for more Quota

LANGUAGE: Python

PERSISTENCE: No persistence

ENVIRONMENT: Pin to original environment (2025-05-...

You won't get new packages, but your code is less likely to break. What is a notebook environment?