Exploring Large Language Models (LLMs) for Text Generation with PyTorch and Hugging Face

Assignment Description:

In this assignment, you will learn and implement Large Language Models (LLMs) for text/code generative purposes using PyTorch and Hugging Face's Transformers library. Through a series of practical exercises, you will gain hands-on experience in fine-tuning pre-trained LLMs, conducting experiments, measuring metrics, exploring parameter variations, and finally documenting your thought process.

1. Fine-Tuning Pre-trained LLMs (30 points): You will begin by fine-tuning pre-trained LLMs. Use at two different types of generative models LLaMa-2(7B) and decoder only model Phi-2 (2.7B) and Mistral (7B). Using PyTorch and Hugging Face's Transformers library, you will implement fine-tuning techniques to adapt the pre-trained model to generate text/code based on an instruction.

For fine-tuning task, you can choose any one of the mentioned datasets:

a) Code Dataset (Python)
b) Instruction Text Dataset

2. Metric Measurements (20 points): After you have trained the model, you will measure various metrics to evaluate the quality of the generated code, including perplexity, BLEU score, ROUGE-L score and BERTSCore and CodeBLEU. You will also do a small-scale human evaluation Table to be generated for this task:

Model Name	BLEU	Rouge-L	BERTScore	CodeBLEU	Human Evaluation (20 Samples)
LLaMA					
Phi-2					
Mistral					

Human Evaluation on Code and Text: For human evaluation on code, you should randomly pick 20 samples of (input instruction, generated code, and ground truth) triplet from your test set. The evaluations should be based on two factors, i) Code Compilability, and ii) Functional Correctness. For each of these factors, put an score between 0-1, and sum up the average.

For all sample i=0 to i=20, Sum($(H_C^i + H_F^i)$ /2)) / 20 ... (1)

For human evaluation on text, you should consider the following factors i) Grammatical correctness, ii) Coherence and iii) Correctness of answer. For this the value should be calculated as:

For all sample i=0 to i=20, Sum($(H_C^i + H_F^i)$ /3)) / 20 ... (2)

Write a discussion (4-5 Lines) explaining the comparison between two models. Moreover, compare the metrics and discuss which metrics are more appropriate compared to human evaluation.

3. Hyperparameter Tuning (15 points): Finally, you will explore the impact of different parameters, top_k, beam_size and temperature, on the text generation capabilities of the fine-tuned LLMs. You will conduct experiments with varying parameter settings and measure their effects on the quality and diversity of the generated text using the metrics defined earlier. Use 4 values for each hyperparameters.

Model Name	Hyperparameters	BLEU	Rouge-L	BERTScore	CodeBLEU	Human Evaluation (20)
LLaMA	Tok_k					
	Beam_size					
	Temperature					
Phi-2	Tok_k					
	Beam_size					
	Temperature					
	Tok_k					
	Beam_size					
	Temperature					

Write another discussion explaining the how the hyperparameters effect on the different metrics of LLaMA and Phi-2 (4-5 Lines).

Deliverables:

Submit your assignment in the form of a GitHub Link with all the instructions in the readme file to run your code. The readme file should follow this GitHub or this GitHub closely.

- Instructions to run the model.

- A document should be uploaded in the GitHub explain the discussions for Task 2 and Task 3.

Note: You may refer to documentation and tutorials available for PyTorch and Hugging Face's Transformers library, but make sure to provide proper citations and/or proper references.