**Summary and Highlights**

Congratulations! You have completed this module. At this point in the course, you know:

* Instruction-tuning involves training models with expert-curated datasets.
* For instruction-tuning, the model requires instructions and answers.
* Instruction-tuning helps perform a wide variety of tasks by interpreting and executing instructions more effectively.
* Instruction-tuning uses three components: Instructions, input, and output.
* Adjust the prompt format to maintain compatibility with the different models’ tokenizers.
* Instruction masking focuses on the loss calculation of specific tokens.
* Load a dataset using CodeAlpaca 20k dataset.
* Format the dataset using the *formating\_prompts\_func* function.
* For creating a formatted dataset, use two code blocks for generating instructions with and without responses.
* To create a model, fine-tune Facebook’s opt-350m model.
* Define the collator using *DataCollatorForCompletionOnlyLM* to prepare data batches for training language models.
* Define the trainer by creating the *SFTTrainer* object.
* Generate a text pipeline from the Transformers library.
* Evaluate the model’s text generation using the BLEU score.
* The reward model takes prompt as an input and response as an output regarding reward or score.
* Reward modeling helps in quantifying quality responses, guiding model optimization, incorporating reward preferences, and ensuring consistency and reliability of the responses.
* The scoring function takes the query and appends the chatbot’s responses.
* The dataset synthetic-instruct-gptj-pairwise from Hugging Face trains and evaluates instruction-following models.
* Defining the *preprocess\_function()* helps format the keys and tokenize the data for the reward trainer.
* *TrainingArguments* class from the Transformers library defines the training arguments.
* Reward trainers orchestrate the process and save and evaluate the model using the *trainer.train()* method.
* The tokenizing process generates scores and compares the output of two functions to achieve the desired win rate.
* Reward model training is an advanced technique that trains a model to identify desired outputs generated by another model and assign scores to the outcome based on its relevance and accuracy.
* Training the scoring function helps generate rewards effectively.
* Generating reward model loss, the encoder model generates responses as contextual embeddings.
* Using the Bradley–Terry reward loss model, you can understand the reward loss model by generating the cost or loss function.