The hyperparameters used in the training process were selected to optimize performance while considering hardware constraints and memory efficiency. The SFTTrainer was configured with a batch size of 2 per device and 4 gradient accumulation steps, effectively simulating a batch size of 8. The learning rate was set to 2e-4, with a linear learning rate scheduler ensuring steady convergence. The training utilized mixed-precision support, with FP16 if bfloat16 was not supported and BF16 otherwise, to optimize memory consumption and computational efficiency. The warmup phase included 5 steps, helping stabilize the initial gradient updates. The evaluation was scheduled every 5 steps, with model checkpoints saved every 5 steps as well, ensuring continuous monitoring and quick recovery if needed. Training logs and evaluation checkpoints were stored in the specified output directory, outputs.

During experimentation, variations in hyperparameters such as larger batch sizes or higher learning rates were tested. Increasing the batch size beyond 2 resulted in memory overflow errors, while higher learning rates above 2e-4 caused training instability and poor convergence. Testing different learning rate schedulers like cosine decay showed slower convergence compared to the linear scheduler. The optimizer choice, adamw\_8bit, was highly effective in balancing memory usage and training speed. This setup provided efficient gradient updates, stable performance, and low memory overhead. The overall configuration ensured robust and reliable model training, while fine-tuning hyperparameters iteratively provided a good balance of performance and hardware efficiency.