Necessary Code TO_BE updated.

```
training_args = TrainingArguments(
    per_device_train_batch_size=1,  # Minimal batch size
    gradient_accumulation_steps=8,  # Accumulate gradients over 8 steps
    ...
)

model.gradient_checkpointing_enable()

from transformers import BitsAndBytesConfig
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True
)
model = AutoModelForCausalLM.from_pretrained("your_model",
quantization_config=bnb_config)

from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained("your_model")
print(tokenizer("Hello world!"))  # Should output clean token IDs, not [UNK] tokens

model_name = "microsoft/phi-2"  # Example
model = AutoModelForCausalLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

Data Formatting & Cleaning: Manual TXT File using plain text, one document per line. Size <= 10 MB (~10K lines of texts). For larger data, pre-process using streaming “datasets” library.

Automatic Cleaning:

```
import re
def clean_text(text):
    text = re.sub(r'[^\w\s.,;!?]', '', text)  # Remove symbols like =====
    text = re.sub(r'\s+', ' ', text)          # Collapse whitespace
    return text.strip()

with open("your_data.txt") as f:
    cleaned_lines = [clean_text(line) for line in f if len(line.split()) > 3]  # Skip short lines

print(cleaned_lines[:10])

training_args = TrainingArguments(
    learning_rate=1e-5,          # Lower LR for stability
    num_train_epochs=2,          # More epochs > more steps
    max_steps=500,               # Hard limit to avoid OOM
    lora_rank=8,                   # Default (higher risks OOM)
    fp16=True,                     # Saves memory
    optim="adamw_torch",           # Default optimizer
)

!nvidia-smi  # Run during training to check memory

# Force coherent outputs
output = model.generate(
    max_new_tokens=50,
    do_sample=True,
    top_k=50,
    top_p=0.95,
    temperature=0.7,
)
```