The Complete SFT Handbook

One Model to Rule Them All: Classification & Merging

1. The Strategy

We are training a single Llama 3.1 8B model to perform two distinct tasks based on the instruction provided:
1. CLASSIFICATION: Analyzing text to generate folder labels/keywords.
2. MERGING: Combining old and new metadata intelligently.

We will upload TWO csv files to Google Colab, combine them, and train the model on the mixed data.

2. The Data Files (Save these locally)

Create a file named 'dataset.csv' with the following content:

```
instruction,input,output
"Analyze the text and classify into JSON. Banned: project, assignment.","In multicellular organisms cells are organised into tissues, organs and organ systems that coordinate specific functions. Evolution explains both the unity and diversity of life.","{""folder_label"": ""Biology"", ""description"": ""cell biology, evolutionary theory, natural selection, homeostasis"", ""keywords"": ""cells, evolution, tissues, biology""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","The rate of a reaction depends on concentration, temperature, presence of catalyst. In aqueous solutions we frequently work with concepts like molarity.","{""folder_label"": ""Chemistry"", ""description"": ""chemical kinetics, solution concentration, thermodynamics"", ""keywords"": ""reaction rate, molarity, chemistry""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Sorting algorithms reorder elements in a collection. Graph algorithms often rely on traversal strategies such as depth first search.","{""folder_label"": ""Computer Science"", ""description"": ""algorithms, data structures, sorting, graph theory, dfs, bfs"", ""keywords"": ""sorting, graphs, programming, optimization""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Basic differentiation rules allow us to compute derivatives. The derivative of a function at a point is defined as the limit of the difference quotient.","{""folder_label"": ""Mathematics"", ""description"": ""calculus, differentiation, limits, derivatives"", ""keywords"": ""calculus, derivatives, math, analysis""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Newton's laws are used as a starting point to connect the net force on an object with its acceleration. Momentum conservation is useful in collisions.","{""folder_label"": ""Physics"", ""description"": ""newtonian mechanics, momentum, force, dynamics"", ""keywords"": ""force, newton, mechanics, physics""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Market capitalization refers to the total dollar market value of a company's outstanding shares.","{""folder_label"": ""Finance"", ""description"": ""stock market, valuation, equity, corporate finance"", ""keywords"": ""market cap, stocks, finance, investing""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Romanticism was an artistic, literary, musical, and intellectual movement that originated in Europe.","{""folder_label"": ""Literature"", ""description"": ""literary movements, 18th century art, european history, emotion"", ""keywords"": ""romanticism, literature, art, history""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Machine learning involves the study of computer algorithms that improve automatically through experience.","{""folder_label"": ""Computer Science"", ""description"": ""artificial intelligence, predictive modeling, neural networks"", ""keywords"": ""machine learning, ai, algorithms""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","Plate tectonics is a scientific theory describing the large-scale motion of plates of the Earth's lithosphere.","{""folder_label"": ""Geology"", ""description"": ""earth science, lithosphere, tectonic plates, geophysics"", ""keywords"": ""plate tectonics, geology, earth""}"
"Analyze the text and classify into JSON. Banned: project, assignment.","The Great Depression was a severe worldwide economic depression that took place mostly during the 1930s.","{""folder_label"": ""History"", ""description"": ""economic history, 1930s, global economy, recession"", ""keywords"": ""great depression, economics, history""}"
```

Create a second file named 'merging_dataset.csv' with the following content:

```
instruction,input,output
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""kinematics, velocity"", ""keywords"": ""motion"" } NEW: { ""description"": ""newton laws, force"", ""keywords"": ""dynamics"" }","{""merged_description"": ""kinematics, velocity, newton laws, force, dynamics"", ""merged_keywords"": ""motion, dynamics,
force""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""cellular respiration"", ""keywords"": ""biology"" } NEW: { ""description"": ""photosynthesis, light energy"", ""keywords"": ""plants"" }","{""merged_description"": ""cellular respiration, photosynthesis, light energy, metabolism"", ""merged_keywords"": ""biology, plants, metabolism""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""algorithms"", ""keywords"": ""coding"" } NEW: { ""description"": ""trees, graphs"", ""keywords"": ""cs"" }","{""merged_description"": ""algorithms, trees, graphs, data structures"", ""merged_keywords"": ""coding, cs, algorithms""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""organic chemistry"", ""keywords"": ""chemistry"" } NEW: { ""description"": ""alkanes, alkenes"", ""keywords"": ""molecules"" }","{""merged_description"": ""organic chemistry, alkanes, alkenes, molecular structure"", ""merged_keywords"": ""chemistry, molecules, organic""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""calculus"", ""keywords"": ""math"" } NEW: { ""description"": ""integrals, area"", ""keywords"": ""calculus"" }","{""merged_description"": ""calculus, derivatives, integrals, area under curve"", ""merged_keywords"": ""math, calculus, area""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""world war 1"", ""keywords"": ""history"" } NEW: { ""description"": ""treaty of versailles"", ""keywords"": ""war"" }","{""merged_description"": ""world war 1, treaty of versailles, 20th century conflict"", ""merged_keywords"": ""history, war, conflict""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""neural networks"", ""keywords"": ""ai"" } NEW: { ""description"": ""deep learning, tensors"", ""keywords"": ""data"" }","{""merged_description"": ""neural networks, deep learning, tensors, ai"", ""merged_keywords"": ""ai, data, deep learning""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""thermodynamics"", ""keywords"": ""heat"" } NEW: { ""description"": ""entropy, carnot cycle"", ""keywords"": ""energy"" }","{""merged_description"": ""thermodynamics, entropy, carnot cycle, thermal physics"", ""merged_keywords"": ""heat, energy, thermodynamics""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""stocks"", ""keywords"": ""finance"" } NEW: { ""description"": ""bonds, risk"", ""keywords"": ""investing"" }","{""merged_description"": ""stocks, bonds, risk management, investment"", ""merged_keywords"": ""finance, investing, stocks""}"
"Merge the following metadata into a single JSON object.","EXISTING: { ""description"": ""poetry"", ""keywords"": ""english"" } NEW: { ""description"": ""sonnets, shakespeare"", ""keywords"": ""literature"" }","{""merged_description"": ""poetry, sonnets, shakespeare, literary analysis"", ""merged_keywords"": ""english, literature, poetry""}"
```

3. The Training Script (Google Colab)

1. Go to colab.research.google.com -> Runtime -> Change runtime -> T4 GPU.
2. Upload BOTH 'dataset.csv' and 'merging_dataset.csv' to the files sidebar.
3. Run this complete script:

```
# === 1. INSTALLATION ===
!pip install "unsloth[colab-new] @ git+https://github.com/unslothai/unsloth.git"
!pip install --no-deps "xformers<0.0.27" "trl<0.9.0" peft accelerate bitsandbytes

import torch
from unsloth import FastLanguageModel
from datasets import load_dataset, concatenate_datasets
from trl import SFTTrainer
from transformers import TrainingArguments

# === 2. MODEL LOADING ===
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit",
    max_seq_length = 2048,
    dtype = None,
    load_in_4bit = True,
)

model = FastLanguageModel.get_peft_model(
    model, r = 16, target_modules = ["q_proj", "k_proj", "v_proj", "o_proj"],
    lora_alpha = 16, lora_dropout = 0, bias = "none",
    use_gradient_checkpointing = "unsloth",
)

# === 3. DATA LOADING (BOTH FILES) ===
# Load classification data
ds_class = load_dataset("csv", data_files="dataset.csv", split="train")
# Load merging data
ds_merge = load_dataset("csv", data_files="merging_dataset.csv", split="train")

# Combine and Shuffle
dataset = concatenate_datasets([ds_class, ds_merge])
dataset = dataset.shuffle(seed=42)

# Format for Alpaca
alpaca_prompt = """### Instruction:
{}

### Input:
{}

### Response:
{}"""

EOS_TOKEN = tokenizer.eos_token
def formatting_prompts_func(examples):
    texts = []
    for instr, inp, out in zip(examples["instruction"], examples["input"], examples["output"]):
        text = alpaca_prompt.format(instr, inp, out) + EOS_TOKEN
        texts.append(text)
    return { "text" : texts, }

dataset = dataset.map(formatting_prompts_func, batched = True)

# === 4. TRAINING ===
trainer = SFTTrainer(
    model = model, tokenizer = tokenizer, train_dataset = dataset,
    dataset_text_field = "text", max_seq_length = 2048,
    args = TrainingArguments(
        per_device_train_batch_size = 2, gradient_accumulation_steps = 4,
        max_steps = 80, # Increased slightly for dual data
        learning_rate = 2e-4, fp16 = not torch.cuda.is_bf16_supported(),
        bf16 = torch.cuda.is_bf16_supported(), logging_steps = 1, output_dir = "outputs",
    ),
)
trainer.train()

# === 5. SAVE ===
model.save_pretrained_gguf("filesense_v1", tokenizer, quantization_method = "q4_k_m")
```

4. Local Setup (Ollama)

1. Download 'filesense_v1-unsloth.Q4_K_M.gguf' from Colab.
2. Create a file named 'Modelfile' (no extension) next to it:

```
FROM ./filesense_v1-unsloth.Q4_K_M.gguf
PARAMETER temperature 0.1
SYSTEM "You are an intelligent file organizer assistant."
```

3. Run command: ollama create filesense -f Modelfile

5. The Python Integration Code

Use this code to make your application talk to your new local model.

```
import requests
import json

def query_ollama(prompt):
    try:
        response = requests.post(
            "http://localhost:11434/api/generate",
            json={
                "model": "filesense",
                "prompt": prompt,
                "format": "json",
                "stream": False
            }
        )
        return json.loads(response.json()['response'])
    except Exception as e:
        print(f"Error: {e}")
        return None

def generate_classification(text):
    prompt = f"""### Instruction:
Analyze the text and classify into JSON. Banned: project, assignment.

### Input:
{text}

### Response:
"""
    return query_ollama(prompt)

def merge_metadata(existing_json, new_json):
    prompt = f"""### Instruction:
Merge the following metadata into a single JSON object.

### Input:
EXISTING: {existing_json} NEW: {new_json}

### Response:
"""
```

```
    return query_ollama(prompt)
```