FineTuning the Multimodel of IDEFICS-9B

1 Download the Required libaries

[ ]:
```
!pip install -q datasets
!pip install -q git+https://github.com/huggingface/transformers
!pip install -q bitsandbytes sentencepiece accelerate loralib
!pip install -q -U git+https://github.com/huggingface/peft.git
```

```
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Building wheel for transformers (pyproject.toml) ... done
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
```

1.0.1 Import the required Libraries

[ ]:
```
import torch
from datasets import load_dataset
from peft import LoraConfig, get_peft_model
from PIL import Image
from transformers import IdeficsForVisionText2Text, AutoProcessor, Trainer,␣
↪TrainingArguments, BitsAndBytesConfig
```

[ ]:
```
device = "cuda" if torch.cuda.is_available() else "cpu"
```

[ ]:
```
checkpoint = "HuggingFaceM4/idefics-9b"
```

2 Quantization Configure

[ ]:
```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
    llm_int8_skip_modules=["lm_head", "embed_tokens"]
)
```

[ ]:
```
processor = AutoProcessor.from_pretrained(checkpoint)
```

```
preprocessor_config.json:   0%|          | 0.00/281 [00:00<?, ?B/s]
tokenizer_config.json:   0%|          | 0.00/1.36k [00:00<?, ?B/s]
tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/1.84M [00:00<?, ?B/s]
added_tokens.json:   0%|          | 0.00/61.0 [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/181 [00:00<?, ?B/s]
Special tokens have been added in the vocabulary, make sure the associated word
embeddings are fine-tuned or trained.
```

[ ]:
```
model = IdeficsForVisionText2Text.from_pretrained(checkpoint,␣
↪quantization_config=bnb_config, device_map="auto")
```

```
config.json:   0%|          | 0.00/1.41k [00:00<?, ?B/s]
model.safetensors.index.json:   0%||  0.00/99.2k [00:00<?, ?B/s]
Downloading shards:   0%|          | 0/19 [00:00<?, ?it/s]
model-00001-of-00019.safetensors:   0%|          | 0.00/2.00G [00:00<?, ?B/s]
model-00002-of-00019.safetensors:   0%|          | 0.00/1.82G [00:00<?, ?B/s]
model-00003-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00004-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00005-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00006-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00007-of-00019.safetensors:   0%|          | 0.00/1.89G [00:00<?, ?B/s]
model-00008-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00009-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00010-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00011-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00012-of-00019.safetensors:   0%|          | 0.00/1.89G [00:00<?, ?B/s]
model-00013-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00014-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00015-of-00019.safetensors:   0%|          | 0.00/1.93G [00:00<?, ?B/s]
model-00016-of-00019.safetensors:   0%|          | 0.00/1.97G [00:00<?, ?B/s]
```

```
model-00017-of-00019.safetensors:   0%|          | 0.00/1.98G [00:00<?, ?B/s]
model-00018-of-00019.safetensors:   0%|          | 0.00/1.97G [00:00<?, ?B/s]
model-00019-of-00019.safetensors:   0%|          | 0.00/705M [00:00<?, ?B/s]
Loading checkpoint shards:0%|          | 0/19 [00:00<?, ?it/s]
generation_config.json:   0%||  0.00/137 [00:00<?, ?B/s]
```

[ ]:
```
model
```

3 Inference

[ ]:
```
# Inference
def do_inference(model, processor, prompts, max_new_tokens=50):
    tokenizer = processor.tokenizer
    bad_words = ["<image>", "<fake_token_around_image>"]
    if len(bad_words) > 0:
        bad_words_ids = tokenizer(bad_words, add_special_tokens=False).input_ids
    eos_token = "</s>"
    eos_token_id = tokenizer.convert_tokens_to_ids(eos_token)

    inputs = processor(prompts, return_tensors="pt").to(device)
    generated_ids = model.generate(
        **inputs,
        eos_token_id=[eos_token_id],
        bad_words_ids=bad_words_ids,
        max_new_tokens=max_new_tokens,
        early_stopping=True
    )
    generated_text = processor.batch_decode(generated_ids,␣
↪skip_special_tokens=True)[0]
    print(generated_text)
```

[ ]:
```
import torchvision.transforms as transforms
```

[ ]:
```
url = "https://hips.hearstapps.com/hmg-prod/images/
↪cute-photos-of-cats-in-grass-1593184777.jpg"
prompts = [
    url,
    "Question: What's on the picture? Answer:",
]
```

4 Preprocessing of dataset

```
[ ]: ##preprocessing
     def convert_to_rgb(image):
       if image.mode == "RGB":
         return image

       image_rgba = image.convert("RGBA")
       background = Image.new("RGBA", image_rgba.size, (255,255,255))
       alpha_composite = Image.alpha_composite(background, image_rgba)
       alpha_composite = alpha_composite.convert("RGB")
       return alpha_composite

     def ds_transforms(example_batch):
       image_size = processor.image_processor.image_size
       image_mean = processor.image_processor.image_mean
       image_std = processor.image_processor.image_std

       image_transform = transforms.Compose([
           convert_to_rgb,
           transforms.RandomResizedCrop((image_size, image_size), scale=(0.9, 1.0),␣
      ↪interpolation=transforms.InterpolationMode.BICUBIC),
           transforms.ToTensor(),
           transforms.Normalize(mean=image_mean, std=image_std)
       ])

       prompts = []
       for i in range(len(example_batch['caption'])):
         caption = example_batch['caption'][i].split(".")[0]
         prompts.append(
             [
                 example_batch['image_url'][i],
                 f"Question: What's on the picture? Answer: This is␣
      ↪{example_batch['name']}. {caption}</s>",
             ],
         )
       inputs = processor(prompts, transform=image_transform, return_tensors="pt").
      ↪to(device)
       inputs["labels"] = inputs["input_ids"]
       return inputs
```

5 Load the dataset

[ ]:
```
#Load and prepare the data
ds = load_dataset("TheFusion21/PokemonCards")
ds = ds["train"].train_test_split(test_size=0.002)
train_ds = ds["train"]
eval_ds = ds["test"]
train_ds.set_transform(ds_transforms)
eval_ds.set_transform(ds_transforms)
```

6 Lora Configuration

[ ]:
```
model_name = checkpoint.split("/")[1]
config = LoraConfig(
    r = 16,
    lora_alpha = 32,
    target_modules = ["q_proj", "k_proj", "v_proj"],
    lora_dropout = 0.05,
    bias="none"
)
```

[ ]:
```
model = get_peft_model(model, config)
```

[ ]:
```
model.print_trainable_parameters()
```

```
trainable params: 19,750,912 || all params: 8,949,430,544 || trainable%:
0.2206946230030432
```

7 Training Arguments

[ ]:
```
training_args = TrainingArguments(
    output_dir = f"{model_name}-PokemonCards",
    learning_rate = 2e-4,
    fp16 = True,
    per_device_train_batch_size = 2,
    per_device_eval_batch_size = 2,
    gradient_accumulation_steps = 8,
    dataloader_pin_memory = False,
    save_total_limit = 3,
    evaluation_strategy ="steps",
    save_strategy = "steps",
    eval_steps = 10,
    save_steps = 25,
    max_steps = 25,
    logging_steps = 5,
    remove_unused_columns = False,
    push_to_hub=False,
    label_names = ["labels"],
    load_best_model_at_end = False,
    report_to = "none",
    optim = "paged_adamw_8bit",
)
```

[ ]:
```
trainer = Trainer(
    model = model,
    args = training_args,
    train_dataset = train_ds,
    eval_dataset = eval_ds
)
```

[ ]:
```
trainer.train()
```

```
<IPython.core.display.HTML object>
```

[ ]:
```
TrainOutput(global_step=25, training_loss=0.7252591323852539,
metrics={'train_runtime': 331.9892, 'train_samples_per_second': 1.205,
'train_steps_per_second': 0.075, 'total_flos': 1942680656828736.0, 'train_loss':
0.7252591323852539, 'epoch': 0.03})
```

[ ]:
```
url = "https://images.pokemontcg.io/pop6/2_hires.png"
```

[ ]:
```
prompts = [
    url,
    "Question: What's on the picture? Answer:",
]
```

[ ]:
```
do_inference(model, processor, prompts, max_new_tokens=100)
```

```
/usr/local/lib/python3.10/dist-
packages/transformers/generation/configuration_utils.py:433: UserWarning:
`num_beams` is set to 1. However, `early_stopping` is set to `True` -- this flag
is only used in beam-based generation modes. You should set `num_beams>1` or
unset `early_stopping`.
  warnings.warn(
```

```
Question: What's on the picture? Answer: This is ['Lucario', 'Gardevoir']. A
Basic Pokemon Card of type Darkness with the title Lucario and 90 HP of rarity
Rare Holo from the set Black Star Promos and the flavor text: It is said that
Lucario is a Pokemon that can only be found in the deepest parts of the forest
```

[ ]:
```
import locale
locale.getpreferredencoding = lambda: "UTF-8"
```

7.1 Push to hub

```
[ ]: huggingface-cli login
```

```
    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|
_|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|          _|
_|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|
_|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|    _|
_|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|    _|
_|    _|    _|_|_|  _|_|_|_|

    To login, `huggingface_hub` requires a token generated from
https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) n
Token is valid (permission: write).
Your token has been saved to /root/.cache/huggingface/token
Login successful
```

```
[ ]: model.push_to_hub(f"{model_name}-PokemonCards", private=False)
```

```
adapter_model.safetensors:   0%|          | 0.00/79.1M [00:00<?, ?B/s]
```

```
[ ]: CommitInfo(commit_url='https://huggingface.co/skuma307/idefics-9b-PokemonCards/c
 ommit/258ff93d25bd52369255a3b339d2a492b3eff3c7', commit_message='Upload model',
 commit_description='', oid='258ff93d25bd52369255a3b339d2a492b3eff3c7',
 pr_url=None, pr_revision=None, pr_num=None)
```

```
[ ]:
```