POC for Llama-v2 For a Basic ChatBbot

Installing required libraries

1. transformers: Hugging face libraries for loading pretrained transformer checkpoints.
2. accelerate : Manages communications between CPU and GPU more efficiently.
3. datasets : For loading datasets from huggingface for future fine tuning.
4. bitsandbytes : Used for quantization of model to run on limited resources.
5. einops :Einstein-Inspired Notation for operations
6. wandb :weights and biases for visualizations

```
!pip install -q -U trl transformers accelerate
!pip install -q datasets bitsandbytes einops wandb
```

```
                                        77.4/77.4 kB 1.1 MB/s eta 0:00:00
                                        7.4/7.4 MB 63.7 MB/s eta 0:00:00
                                        244.2/244.2 kB 21.3 MB/s eta 0:00:00
                                        486.2/486.2 kB 30.0 MB/s eta 0:00:00
                                        268.8/268.8 kB 16.9 MB/s eta 0:00:00
                                        7.8/7.8 MB 81.0 MB/s eta 0:00:00
                                        1.3/1.3 MB 52.4 MB/s eta 0:00:00
                                        110.5/110.5 kB 12.1 MB/s eta 0:00:00
                                        212.5/212.5 kB 13.9 MB/s eta 0:00:00
                                        134.8/134.8 kB 11.4 MB/s eta 0:00:00
                                        134.3/134.3 kB 7.8 MB/s eta 0:00:00
                                        92.6/92.6 MB 9.9 MB/s eta 0:00:00
                                        42.2/42.2 kB 4.2 MB/s eta 0:00:00
                                        2.1/2.1 MB 71.9 MB/s eta 0:00:00
                                        188.5/188.5 kB 14.0 MB/s eta 0:00:00
                                        214.7/214.7 kB 17.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
                                        62.7/62.7 kB 2.5 MB/s eta 0:00:00
  Building wheel for pathtools (setup.py) ... done
```

```
from huggingface_hub import login
login()
```

```
Token is valid (permission: read).
ı has been saved in your configured git credential helpe
ır token has been saved to /root/.cache/huggingface/tol
Login successful
```

Imports

1. Pipeline : High level pipeline to interact with huggingface models
2. torch : Deeplearning framework used
3. transformers: Huggingface library to interact with transformer models.

```
# Use a pipeline as a high-level helper
from transformers import pipeline
import torch
import transformers
```

BY DIRECTLY USING PIELINE

```
model="meta-llama/Llama-2-7b-chat-hf"
pipeline = transformers.pipeline(
    "text-generation",
    model=model,
    torch_dtype=torch.float16)
```

From transformers import

1. AutoModelForCausalLM : Instantiates one of the model classes of the library (with a causal language modeling head) from a configuration.
2. AutoTokenizer : Tokenizer for a selected model
3. BitsAndBytesConfig : returns object which is used to change datatype / quantise a model

```
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig

model_name = "meta-llama/Llama-2-7b-chat-hf"

bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
)

model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
    trust_remote_code=True
)
model.config.use_cache = False
```

```
Downloading (…)lve/main/config.json: 100%  635/635 [00:00<00:00, 43.1kB/s]
Downloading (…)fetensors.index.json: 100%  26.8k/26.8k [00:00<00:00, 1.23MB/s]
Downloading shards: 100%  2/2 [01:14<00:00, 34.74s/it]
Downloading (…)of-00002.safetensors: 100%  9.98G/9.98G [00:51<00:00, 294MB/s]
Downloading (…)of-  3.50G/3.50G [00:22<00:00,
```

```
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
```

```
Downloading (…)okenizer_config.json: 100%  770/770 [00:00<00:00, 40.0kB/s]
Downloading tokenizer.model: 100%  500k/500k [00:00<00:00, 21.8MB/s]
Downloading  1.84M/1.84M [00:00<00:00
```

Testing the chat model

Steps:

1. Prompt : list of prompts to be passed to model
2. inputs : tokenization of prompt

   2.1 adding padding =True if multiple input sequences are of different lenghts
3. generate_ids: gnerated ids of model for a given list of prompts
4. using tokenizer.batch_decode to decode the genrated ids into human readable sentences.

```
prompt = ["List all the types of melons.",'How to quickly breathe?']
inputs = tokenizer(prompt, return_tensors="pt",padding =True)
generate_ids = model.generate(inputs.input_ids, max_length=500)
tokenizer.batch_decode(generate_ids, skip_special_tokens=True, clean_up_tokenization_spaces=False)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1468: Use
  warnings.warn(
"List all the types of melons. Unterscheidung between the different types of melo
ns.\n\nThere are several types of melons, including:\n\n1. Cantaloupe: This is th
e most common type of melon, with a smooth, yellow skin and sweet, juicy flesh.\n
2. Honeydew: Similar to cantaloupe, but with a slightly different flavor and text
ure. Honeydew melons have a more pronounced sweetness and a slightly firmer fles
h.\n3. Watermelon: Watermelon is a large, round melon with a green or yellow skin
```

```
tokenizer.batch_decode(generate_ids, skip_special_tokens=True, clean_up_tokenization_spaces=False)[1]
```

```
"How to quickly breathe?\n\nBreathing is a vital function that helps to bring oxy
gen into the body and remove carbon dioxide. Here are some tips on how to quickly
breathe:\n\n1. Take Deep Breaths: When you\'re feeling stressed or anxious, your
breathing can become shallow and rapid. To calm down, try taking deep breaths thr
ough your nose and exhaling through your mouth. Inhale for a count of four, hold
your breath for a count of four, and exhale for a count of four.\n\n2. Practice D
```

How to quickly breathe?

Breathing is a vital function that helps to bring oxygen into the body and remove carbon dioxide. Here are some tips on how to quickly breathe:

1. Take Deep Breaths: When you're feeling stressed or anxious, your breathing can become shallow and rapid. To calm down, try taking deep breaths through your nose and exhaling through your mouth. Inhale for a count of four, hold your breath for a count of four, and exhale for a count of four.
2. Practice Diaphragmatic Breathing: The diaphragm is a muscle that helps you breathe more efficiently. To practice diaphragmatic breathing, place one hand on your stomach and the other on your chest. Inhale through your nose, allowing your stomach to rise as your diaphragm descends. Exhale through your mouth, allowing your stomach to fall as your diaphragm rises.
3. Use Your Vocal Cords: When you're feeling anxious or stressed, try using your vocal cords to help you breathe more deeply. Take a deep breath in through your nose, and then exhale slowly through your mouth, making a "hmm" sound. This can help you relax and focus on your breathing.
4. Try Box Breathing: Box breathing is a simple technique that can help you breathe more deeply and evenly. To practice box breathing, inhale for a count of four, hold your breath for a count of four, exhale for a count of four, and then hold your breath again for a count of four. Repeat this pattern several times.
5. Practice Progressive Muscle Relaxation: Progressive muscle relaxation is a technique that can help you relax and focus on your breathing. To practice progressive muscle relaxation, start by tensing and relaxing different muscle groups in your body, such as your toes, calves, and fingers. As you tense each muscle group, hold your breath for a count of four, and then release and exhale slowly.
6. Use Visualization Techniques: Visualization can help you