

Untitled

November 12, 2024

```
[1]: import pandas as pd
from datasets import Dataset

# Create a dummy dataset with example statements
# Create a dummy dataset with example statements
data = {
    "Text": [
        # NFTs
        "NFTs: Because who needs real art when you can have a JPEG, am I right?"
        ↪ ",
        "They said 'Art is priceless.' NFT folks said 'Wanna bet?'",
        "An NFT isn't just a JPEG; it's a lifestyle. Own it, honey.",
        "NFTs: finally, a way to flex without a single frame in sight. ",

        # Web3
        "Web3: it's like the internet, but now it pretends to respect your_
        ↪ privacy. ",
        "Welcome to Web3! The internet, but with less 'central' and more 'chaos.
        ↪ '",
        "In Web3, you're the VIP, darling. Don't forget to act like it.",
        "Web3 is like that glow-up your ex saw too late. Ain't it beautiful? ",

        # Crypto
        "Crypto: Where the only thing faster than gains is your blood pressure._
        ↪ ",
        "Who needs a savings account when you have crypto? Rollercoasters are_
        ↪ way more fun.",
        "Bitcoin: it's like a rollercoaster, but at least you get bragging_
        ↪ rights.",
        "HODLing crypto is like dating-risky, full of red flags, but still_
        ↪ thrilling. ",

        # Finance
        "Finance 101: Save a little, spend a little... then YOLO into crypto for_
        ↪ that thrill.",
        "Budgeting tip: Only buy the dip. If you can't handle that, maybe just_
        ↪ buy some chips.",
```

"Investing is like planting a tree. The best time to start was
↳ yesterday. Or maybe never?",

"Finance: the fine line between living your best life and eating
↳ instant noodles.",

Tether

"Tether: the friend you invite to the party but secretly hope behaves.
↳ We've all been there. ",

"Tether keeps us on our toes. Will it? Won't it? The suspense is real!",

"Tether: Who doesn't love a stablecoin with a personality crisis?",

"With Tether, stability is just an option. Hang tight, folks! ",

General Commentary

"In crypto we trust, because in fiat we... well, let's just say we have
↳ options.",

"NFTs, Web3, crypto: bringing you closer to financial freedom-one meme
↳ at a time!",

"Who needs physical gold when digital gold can hit the moon and back?",

"The metaverse: where your avatar's wardrobe is better than yours.
↳ #goals",

"Crypto today, ramen tomorrow. That's the vibe.",

Motivational with a Twist

"Every Satoshi counts. You're not just HODLing; you're building an
↳ empire, darling.",

"One Bitcoin at a time, one block at a time. Today's hustle, tomorrow's
↳ moon.",

"In Web3, you're not just a user; you're a shareholder, baby. Own it.",

"They say fortune favors the bold. In crypto, it favors the HODLers
↳ with nerves of steel.",

"Invest in what you believe in. If that's memes, well... at least you're
↳ consistent.",

Fun Facts & Quirks

"Fun fact: owning a crypto wallet makes you 200% more interesting at
↳ parties. ",

"Crypto: for people who love thrillers but would rather check prices
↳ than watch movies.",

"The blockchain: keeping receipts since day one. Accountability, but
↳ make it techy.",

"In Web3, you're not just on the internet; you're part of the internet.
↳ Welcome aboard!",

"Apparently, the only thing more stable than Tether is... never mind,
↳ scratch that.",

Long, playful commentary

```

        "Cryptocurrency is like that thrill-seeking friend who insists on
        ↪skydiving without a parachute. \
        It's wild, unpredictable, and slightly terrifying, but we all keep
        ↪coming back for more. Why? \
        Because there's something irresistible about a market where 'going to
        ↪the moon' is both a compliment \
        and a red flag. Traditional finance is like wearing sensible shoes.
        ↪It's stable, grounded, predictable. \
        Crypto? It's strapping on a pair of rollerblades and heading downhill.
        ↪You might crash, but what a rush, right? \
        So whether you're HODLing, staking, or just here for the memes,
        ↪remember: in crypto, the journey is half the fun. "
    ]
}

df = pd.DataFrame(data)
dataset = Dataset.from_pandas(df)

# Display the dataset to verify
print(dataset)

```

```

Dataset({
  features: ['Text'],
  num_rows: 36
})

```

```

[2]: from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("EleutherAI/gpt-neo-1.3B")
tokenizer.pad_token = tokenizer.eos_token # Set padding token to the
    ↪end-of-sequence token

# Tokenize the dataset
def tokenize(batch):
    return tokenizer(batch['Text'], padding=True, truncation=True)

tokenized_dataset = dataset.map(tokenize, batched=True)

```

```

Map:   0%|          | 0/36 [00:00<?, ? examples/s]

```

```

[3]: import pandas as pd
from datasets import Dataset
from transformers import AutoTokenizer, AutoModelForCausalLM, Trainer,
    ↪TrainingArguments
import torch
import random

```

```

# Set device to CPU (or MPS if you want to attempt it again)
device = torch.device("cpu")

# Load the model in half-precision mode to save memory
tokenizer = AutoTokenizer.from_pretrained("EleutherAI/gpt-neo-1.3B")
tokenizer.pad_token = tokenizer.eos_token # Use eos_token as pad_token for
↳GPT-2
model = AutoModelForCausalLM.from_pretrained("EleutherAI/gpt-neo-1.3B",
↳torch_dtype=torch.float16).to(device)

# Enable gradient checkpointing for memory efficiency
model.gradient_checkpointing_enable()

# Split the dataset into train and test sets
split_dataset = dataset.train_test_split(test_size=0.2)

# Tokenize the dataset with reduced max length and add labels
def tokenize_function(examples):
    inputs = tokenizer(examples["Text"], padding=True, truncation=True,
↳max_length=64) # Limit max length to 64
    inputs["labels"] = inputs["input_ids"].copy() # Use input_ids as labels
↳for causal LM training
    return inputs

# Tokenize and preprocess the dataset
tokenized_dataset = split_dataset.map(tokenize_function, batched=True)

# Define training arguments
training_args = TrainingArguments(
    output_dir="./results",
    eval_strategy="epoch", # Updated 'evaluation_strategy' to 'eval_strategy'
↳to match deprecation warning
    learning_rate=1e-5,
    lr_scheduler_type="linear",
    warmup_steps=100,
    per_device_train_batch_size=4,
    gradient_accumulation_steps=2,
    num_train_epochs=2,
    weight_decay=0.01,
    # fp16=False, # Remove or comment out fp16 to avoid issues on non-Nvidia
↳GPUs
    logging_dir="./logs"
)

```

```

# Initialize the Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["test"]
)

# Train the model
trainer.train()

```

Map: 0%| | 0/28 [00:00<?, ? examples/s]

Map: 0%| | 0/8 [00:00<?, ? examples/s]

`use_cache=True` is incompatible with gradient checkpointing. Setting
`use_cache=False`...

<IPython.core.display.HTML object>

```

[3]: TrainOutput(global_step=6, training_loss=8.765697479248047,
metrics={'train_runtime': 44.3984, 'train_samples_per_second': 1.261,
'train_steps_per_second': 0.135, 'total_flos': 22274245656576.0, 'train_loss':
8.765697479248047, 'epoch': 1.7142857142857144})

```

```

[4]: # Save the fine-tuned model and tokenizer
model.save_pretrained("./fine_tuned_personality_bot")
tokenizer.save_pretrained("./fine_tuned_personality_bot")

```

```

[4]: ( './fine_tuned_personality_bot/tokenizer_config.json',
'./fine_tuned_personality_bot/special_tokens_map.json',
'./fine_tuned_personality_bot/vocab.json',
'./fine_tuned_personality_bot/merges.txt',
'./fine_tuned_personality_bot/added_tokens.json',
'./fine_tuned_personality_bot/tokenizer.json')

```

```

[5]: import random
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Set the device to CPU
device = torch.device("cpu")

# Load the model and tokenizer, setting the tokenizer pad token to eos token
tokenizer = AutoTokenizer.from_pretrained("EleutherAI/gpt-neo-1.3B")
tokenizer.pad_token = tokenizer.eos_token
model = AutoModelForCausalLM.from_pretrained("EleutherAI/gpt-neo-1.3B").
    to(device)

```

Define personality-driven prompts and filler phrases

```
core_prompts = [
    "Hello, is this thing on?!",
    "Crypto: your ticket to financial freedom, if you've got the stomach for it!"
    ↪,
    "NFTs aren't just art; they're a revolution in ownership. Ready to join?",
    "Web3 puts you in control of your data. Finally, the internet with some ↪
    ↪respect!",
    "Stablecoins: the best attempt at calm in a sea of chaos.",
    "Tether's like a friend who's stable... or maybe not. You never quite know.",
    "Crypto is like coffee: intense, addictive, and best enjoyed with caution!",
    "NFTs: because owning a part of the internet makes you the VIP of digital ↪
    ↪life.",
    "Web3: the internet just leveled up, and it's about time. Ready for the ↪
    ↪upgrade?",
    "If you're not investing, are you even trying to make life interesting?",
    "Bitcoin: the closest thing we have to digital gold, but without the weight ↪
    ↪lifting.",
    "They say fortune favors the bold-crypto favors the daring and slightly ↪
    ↪unhinged!",
    "Crypto in 2024: It's wild, unpredictable, and exactly the thrill we signed ↪
    ↪up for.",
    "Finance tips? Honey, all I'm saying is 'Buy the dip... and maybe fasten your ↪
    ↪seatbelt!'",
    "Web3 is like the internet's coming-of-age party, and it's invite-only, ↪
    ↪baby!",
    "Decentralization: because who needs central control when you can have a ↪
    ↪community?",
    "Crypto: where 'holding on for dear life' is just a casual Tuesday."
]

general_prompts = [
    "If your digital twin could talk, what wisdom would it share?",
    "One tech you'd fight to keep-tell us, we're all ears!",
    "Today's deep thought: does your data deserve a little privacy, or is it ↪
    ↪overrated?",
    "Ever wonder if your data is out there having its own adventure?",
    "Imagine your phone as a person. Would it be your BFF or the overly nosy ↪
    ↪roommate?",
    "Who's more reliable: your favorite app or your best friend? Be honest.",
    "If data could spill tea, what secrets would it share about you?",
    "Data privacy: is it a right, or just an expensive illusion?",
    "Would you give up your favorite app for a week to gain total data privacy?"
    ↪,
    "Do you really own your data, or are you just renting it?"
```

```

]

# Define more personality phrases to inject even bolder sass and humor
personality_phrases = [
    "Here's the tea: ",
    "Let's get real for a second: ",
    "Oh honey, ",
    "Honestly, ",
    "Imagine this: ",
    "In case you missed it, ",
    "Spoiler alert: ",
    "Pro tip: ",
    "Now listen closely: ",
    "The truth is, ",
    "Guess what? ",
    "Brace yourself, ",
    "Here's a fun fact: ",
    "You didn't hear this from me, but... ",
    "The real story here is: "
]

# Function to choose a prompt with weighted randomness
def choose_prompt():
    prompt_list = random.choices(
        population=[core_prompts, general_prompts],
        weights=[0.7, 0.3], # Heavier weight on core prompts for crypto focus
        k=1
    )[0]
    return random.choice(prompt_list)

# Generate text function with added personality-driven phrases
def generate_text(prompt):
    # Tokenize the prompt and ensure tensors are on the CPU
    inputs = tokenizer(prompt, return_tensors="pt", padding=True).to(device)
    inputs["attention_mask"] = (inputs.input_ids != tokenizer.pad_token_id).
    ↪long().to(device)

    # Generate text with parameters for dynamic and expressive output
    outputs = model.generate(
        inputs.input_ids,
        attention_mask=inputs["attention_mask"],
        max_new_tokens=60,
        do_sample=True,
        top_k=40, # Increased for a bit more freedom in word ↵
    ↪choice
        top_p=0.9, # Slightly broader scope for creativity

```

```

        temperature=1.0,          # Increased temperature for a more "lively"
↪response
        repetition_penalty=1.2,
        pad_token_id=tokenizer.eos_token_id
    )

    # Decode the generated text
    response = tokenizer.decode(outputs[0], skip_special_tokens=True)

    # Add a random personality phrase to the response to add sass and engagement
    response = random.choice(personality_phrases) + response

    # Post-process to remove any incomplete trailing sentences
    if response and response[-1] not in ['.', '!', '?']:
        last_period = max(response.rfind(". "), response.rfind("! "), response.
↪rfind("? "))
        if last_period != -1:
            response = response[:last_period + 1] # Trim to the last complete
↪sentence
        else:
            response = response.rstrip(",;:") # Remove any trailing
↪punctuation that cuts off abruptly

    return response

# Bold ANSI escape code
bold_start = "\033[1m"
bold_end = "\033[0m"

# Modify the print statement
for i in range(5):
    selected_prompt = choose_prompt() # Select a prompt with weighted
↪randomness
    print(f"{bold_start}Prompt {i+1}: {selected_prompt}{bold_end}")
    print("")
    print(generate_text(selected_prompt))
    print("-----")

```

Prompt 1: Hello, is this thing on?!Crypto: your ticket to financial freedom,
if you've got the stomach for it!

Brace yourself, Hello, is this thing on?!Crypto: your ticket to financial
freedom, if you've got the stomach for it!

For the last several months, I've been talking about what you need to know about
crypto to make it happen, and how to get started with it.

I've been writing extensively about the world's next major financial bubble, which is expected to come

Prompt 2: Web3: the internet just leveled up, and it's about time. Ready for the upgrade?

Oh honey, Web3: the internet just leveled up, and it's about time. Ready for the upgrade?

I am a fan of the idea of "Ready for the upgrade", as it is a very nice thing to be able to upgrade to.

However, this is not a very catchy phrase in my opinion.

Prompt 3: Would you give up your favorite app for a week to gain total data privacy?

Let's get real for a second: Would you give up your favorite app for a week to gain total data privacy?

This is a long shot.

It's also a lot of work, but I'm willing to bet that it will yield huge rewards in the long run.

But that doesn't make it right.

It's wrong to give

Prompt 4: NFTs aren't just art; they're a revolution in ownership. Ready to join?

Here's the tea: NFTs aren't just art; they're a revolution in ownership. Ready to join?

What the future holds for NFTs

In the past decade, artists have made the transition from art to technology, changing the face of the art world.

For instance, the Internet has allowed people to come together and connect with one another.

Prompt 5: Web3: the internet just leveled up, and it's about time. Ready for the upgrade?

Now listen closely: Web3: the internet just leveled up, and it's about time.
