Untitled

November 12, 2024

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

# Create a dummy dataset with example statements
data = {
    "Text": [
        # NFTs
        "NFTs: Because art was getting a bit too traditional for us. Now it's␣
↪digital and fabulous! ",
        "They said 'You can't own the internet'-NFT folks said 'Challenge␣
↪accepted!' ",
        "An NFT isn't just a JPEG; it's a lifestyle choice. Welcome to the␣
↪club, honey.",
        "NFTs: finally, a way to flex your art collection without a single␣
↪frame. ",
        "Owning a digital collectible? That's the new art of collecting. A␣
↪Picasso in your pocket? Yes, please!",

        # Web3
        "Web3: It's like the internet, but with less 'central' and more 'magic.
↪' ",
        "Welcome to Web3! Where you can be a part-owner of everything and␣
↪nothing all at once.",
        "In Web3, your data finally gets the respect it deserves. Privacy? Oh,␣
↪we fancy that.",
        "Web3: where you, the user, are finally the VIP, and the internet knows␣
↪it. ",
        "Some think Web3 is hype. I think it's just the internet's glow-up␣
↪moment.",

        # Crypto
        "Crypto: the only asset where 'going to the moon' is a real thing. ",
        "Why have a savings account when you can have a rollercoaster called␣
↪crypto?",
        "Bitcoin isn't just currency; it's the thrill ride we all signed up for.
↪ ",
```

```
        "Holding crypto is like a first date: risky but with major potential. ",
        "Crypto in 2024: volatile, unpredictable, and totally our kind of party.",

        # Finance
        "Finance 101: save some, spend some, invest some… and maybe sprinkle in a bit of crypto?",
        "Budgeting tip: Only buy the dip. And if you can't handle the dip, maybe just get the chips.",
        "In finance, like in life, it’s not about timing the market; it’s about time *in* the market.",
        "Investing? Think of it like planting a tree. The best time to start was 10 years ago, the second best time is today.",
        "Finance is the fine balance between spending for joy and saving for freedom. Nail it, and you’re golden.",

        # Tether
        "Tether: the friend you invite to the party, but secretly hope behaves this time.  ",
        "Why get a financial cushion when you have Tether? It’ll keep you guessing!",
        "Tether keeps us on our toes. Will it, won’t it? Ah, the suspense!",
        "Tether: because who doesn’t love a good plot twist in the world of stablecoins?",
        "With Tether, you get stability... or at least the thrill of the promise of it. Hang tight!  ",

        # General commentary
        "In crypto we trust, because in fiat we… well, let’s just say we like variety.",
        "NFTs, Web3, and crypto: bringing you closer to financial freedom-one meme at a time!",
        "Who needs physical gold when digital gold can hit the moon and beyond?",
        "The metaverse: where your avatars have a better wardrobe than you. Talk about goals!",
        "In the world of crypto, today's volatility is tomorrow’s dinner conversation.",

        # Motivational with a twist
        "Remember, every Satoshi counts. You’re not just HODLing; you’re building an empire!",
        "One Bitcoin at a time, one block at a time. Today’s grind, tomorrow’s moonshot.",
```

```
        "In Web3, you're not just a user; you're a stakeholder. Own your world,␣
↪baby.",
        "They say fortune favors the brave. In crypto, it also favors the␣
↪HODLers!",
        "Invest in what you believe in. And if you believe in memes, well,␣
↪there's always Dogecoin.",

        # Fun facts & quirks
        "Fun fact: owning a crypto wallet makes you 200% more interesting at␣
↪parties.  ",
        "Crypto: for people who love thrillers but prefer checking prices over␣
↪watching movies.",
        "The blockchain: keeping receipts since day one. Accountability has␣
↪never looked so 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… well, let's␣
↪just say, stay tuned!",

        "NFTs: Because art was getting a bit too traditional for us. Now it's␣
↪digital and fabulous!  ",
        "Web3: It's like the internet, but with less 'central' and more 'magic.
↪'  ",
        "Crypto isn't just a currency-it's a lifestyle choice. Ready to dive in?
↪",
        "The future is decentralized, darling. Here's what Web3 has in store!",
        "Bitcoin isn't just currency; it's the thrill ride we all signed up for.
↪  ",
        "Money matters, but how you spend it matters more. Here's a quirky view␣
↪on finance:",
        "Tether: the friend you invite to the party but secretly hope behaves.␣
↪",

    ]
}

# Convert the dictionary to a DataFrame
df = pd.DataFrame(data)

# Convert the DataFrame to a Hugging Face Dataset
dataset = Dataset.from_pandas(df)

# Display the dataset to verify
print(dataset)
```

```
Dataset({
    features: ['Text'],
```

```
    num_rows: 47
})
```

[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%|          | 0/47 [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()

# Create a dummy dataset with example statements
data = {
    "Text": [
        # NFTs
        "NFTs: Because art was getting a bit too traditional for us. Now it's␣
 ↪digital and fabulous!  ",
        "They said 'You can't own the internet'-NFT folks said 'Challenge␣
 ↪accepted!'  ",
        "An NFT isn't just a JPEG; it's a lifestyle choice. Welcome to the␣
 ↪club, honey.",
```

```
        "NFTs: finally, a way to flex your art collection without a single␣
↪frame.  ",
        "Owning a digital collectible? That's the new art of collecting. A␣
↪Picasso in your pocket? Yes, please!",

        # Web3
        "Web3: It's like the internet, but with less 'central' and more 'magic.
↪'  ",
        "Welcome to Web3! Where you can be a part-owner of everything and␣
↪nothing all at once.",
        "In Web3, your data finally gets the respect it deserves. Privacy? Oh,␣
↪we fancy that.",
        "Web3: where you, the user, are finally the VIP, and the internet knows␣
↪it.  ",
        "Some think Web3 is hype. I think it's just the internet's glow-up␣
↪moment.",

        # Crypto
        "Crypto: the only asset where 'going to the moon' is a real thing.  ",
        "Why have a savings account when you can have a rollercoaster called␣
↪crypto?",
        "Bitcoin isn't just currency; it's the thrill ride we all signed up for.
↪  ",
        "Holding crypto is like a first date: risky but with major potential.␣
↪ ",
        "Crypto in 2024: volatile, unpredictable, and totally our kind of party.
↪",

        # Finance
        "Finance 101: save some, spend some, invest some… and maybe sprinkle in␣
↪a bit of crypto?",
        "Budgeting tip: Only buy the dip. And if you can't handle the dip,␣
↪maybe just get the chips.",
        "In finance, like in life, it's not about timing the market; it's about␣
↪time *in* the market.",
        "Investing? Think of it like planting a tree. The best time to start␣
↪was 10 years ago, the second best time is today.",
        "Finance is the fine balance between spending for joy and saving for␣
↪freedom. Nail it, and you're golden.",

        # Tether
        "Tether: the friend you invite to the party, but secretly hope behaves␣
↪this time.  ",
        "Why get a financial cushion when you have Tether? It'll keep you␣
↪guessing!",
        "Tether keeps us on our toes. Will it, won't it? Ah, the suspense!",
```

```
        "Tether: because who doesn't love a good plot twist in the world of␣
↪stablecoins?",
        "With Tether, you get stability... or at least the thrill of the␣
↪promise of it. Hang tight! ",

        # General commentary
        "In crypto we trust, because in fiat we… well, let's just say we like␣
↪variety.",
        "NFTs, Web3, and crypto: bringing you closer to financial freedom-one␣
↪meme at a time!",
        "Who needs physical gold when digital gold can hit the moon and beyond?
↪",
        "The metaverse: where your avatars have a better wardrobe than you.␣
↪Talk about goals!",
        "In the world of crypto, today's volatility is tomorrow's dinner␣
↪conversation.",

        # Motivational with a twist
        "Remember, every Satoshi counts. You're not just HODLing; you're␣
↪building an empire!",
        "One Bitcoin at a time, one block at a time. Today's grind, tomorrow's␣
↪moonshot.",
        "In Web3, you're not just a user; you're a stakeholder. Own your world,␣
↪baby.",
        "They say fortune favors the brave. In crypto, it also favors the␣
↪HODLers!",
        "Invest in what you believe in. And if you believe in memes, well,␣
↪there's always Dogecoin.",

        # Fun facts & quirks
        "Fun fact: owning a crypto wallet makes you 200% more interesting at␣
↪parties. ",
        "Crypto: for people who love thrillers but prefer checking prices over␣
↪watching movies.",
        "The blockchain: keeping receipts since day one. Accountability has␣
↪never looked so 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… well, let's␣
↪just say, stay tuned!",

        "NFTs: Because art was getting a bit too traditional for us. Now it's␣
↪digital and fabulous! ",
        "Web3: It's like the internet, but with less 'central' and more 'magic.
↪' ",
```

```
        "Crypto isn't just a currency-it's a lifestyle choice. Ready to dive in?
 ↪",
        "The future is decentralized, darling. Here's what Web3 has in store!",
        "Bitcoin isn't just currency; it's the thrill ride we all signed up for.
 ↪  ",
        "Money matters, but how you spend it matters more. Here's a quirky view␣
 ↪on finance:",
        "Tether: the friend you invite to the party but secretly hope behaves.␣
 ↪ ",

    ]
}

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

# 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",
    evaluation_strategy="no",  # No evaluation dataset for this example
    learning_rate=2e-5,
    per_device_train_batch_size=1,
    gradient_accumulation_steps=4,  # Further reduce accumulation steps if␣
 ↪needed
    num_train_epochs=0.3,  # Reduced epochs for memory efficiency
    weight_decay=0.01,
    fp16=False,
    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%|          | 0/37 [00:00<?, ? examples/s]
```

```
Map:   0%|          | 0/10 [00:00<?, ? examples/s]
```

```
/opt/anaconda3/envs/tbot/lib/python3.12/site-
packages/transformers/training_args.py:1568: FutureWarning:
`evaluation_strategy` is deprecated and will be removed in version 4.46 of
Transformers. Use `eval_strategy` instead
  warnings.warn(
`use_cache=True` is incompatible with gradient checkpointing. Setting
`use_cache=False`…
```

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

[3]:
```
TrainOutput(global_step=3, training_loss=21.377604166666668,
metrics={'train_runtime': 21.4391, 'train_samples_per_second': 0.518,
'train_steps_per_second': 0.14, 'total_flos': 2523254390784.0, 'train_loss':
21.377604166666668, 'epoch': 0.32432432432432434})
```

[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, and set 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 core and general prompts
core_prompts = [
    "NFTs aren't just art; they're a movement. Why?",
    "In Web3, you're the VIP. But what does that mean?",
    "Crypto's not just currency. It's a wild ride. Here's why:",
    "Stablecoins: revolutionary or risky?",
    "You think NFTs are just JPEGs? Let's chat about ownership."
]

general_prompts = [
    "If your digital twin could speak, what would it say?",
    "What's one tech innovation we can't live without?",
    "Today's thought: is privacy even possible online?",
    "Ever thought about what your data is doing right now?",
    "Imagine your phone is a person-how would you two get along?",
    "What is on your mind?"
]

# Function to choose a prompt with weighted randomness
def choose_prompt():
    prompt_list = random.choices(
        population=[core_prompts, general_prompts],
        weights=[0.5, 0.5],  # 50% chance for each category
        k=1
    )[0]
    return random.choice(prompt_list)  # Randomly select a prompt within the␣
 ↪chosen category

# Generate text function with adjusted sampling parameters
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 using the model with modified parameters
    outputs = model.generate(
        inputs.input_ids,
        attention_mask=inputs["attention_mask"],
        max_length=100,  # Limit length for conciseness
        do_sample=True,
        top_k=30,        # More focused sampling
        top_p=0.8,       # Less diverse sampling
        temperature=1.2, # Increase creativity
```

```
        pad_token_id=tokenizer.eos_token_id
    )

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

# Generate and print responses for 5 different prompts
for i in range(5):
    selected_prompt = choose_prompt()  # Select a prompt with weighted␣
 ↪randomness
    print(f"Prompt {i+1}: {selected_prompt}")
    print(generate_text(selected_prompt))
    print("------------")
```

```
Prompt 1: If your digital twin could speak, what would it say?
If your digital twin could speak, what would it say?

It might not be much, but your digital twin, a computer simulation of you, could
answer. The computer simulation would need to be accurate to be useful,
though-it would have to mimic what you actually do when you're awake. And it
would be far more useful if it knew about all the ways you have a mind-and how
those thoughts affect your actions.

If you've been reading this column for a
------------
Prompt 2: What’s one tech innovation we can’t live without?
What’s one tech innovation we can’t live without? If so, it might be the latest
and greatest gadget, the next Google product, the iPhone. And while it would be
hard to argue with that claim, we think that it may not have anything to do with
that new device. That is until we meet the people behind Google Wave, the
company’s most recent attempt at making a new product that actually exists.

What Is Wave?

As the
------------
Prompt 3: NFTs aren’t just art; they’re a movement. Why?
NFTs aren’t just art; they’re a movement. Why? Because it’s time we were talking
about it. Because if you can’t talk about what we need to do to fix these
problems, how will you know you’ve even solved them?

You don’t know until you try. The problem we’ve seen since September 11 is that
the United States and its allies in the world, like the U.
------------
Prompt 4: Imagine your phone is a person-how would you two get along?
Imagine your phone is a person-how would you two get along? How would your
family be if you lived in the city? How would your boss react to your idea to
```

move? And the answer to these questions are, "It’s none of your business."

But, how would it feel if you were a person, not a phone? Would your family and friends care? Would they be angry and frustrated? Would they even like your move?

And

Prompt 5: Stablecoins: revolutionary or risky?
Stablecoins: revolutionary or risky?

By James W. Wetherbee

Dec. 21, 2017, 12:45 p.m.

The future of money is in the stars. In the early 21st century, stars will be in our pockets. They will not come to our desks, but they will shine bright as they move in our vision. The new money of the future will not come from governments or banks. Instead, they will come from the heavens themselves
---

[ ]:

[ ]: