Amardeep Singh

E23CSEU2189

```
# ================= IMPORTS =================
import tensorflow as tf
import numpy as np

# ================= LOAD DATASET FROM FILE =================
with open("sample text.txt", "r", encoding="utf-8") as f:
    text = f.read().lower()

# ================= PREPROCESSING =================
chars = sorted(list(set(text)))
vocab_size = len(chars)

char_to_idx = {c: i for i, c in enumerate(chars)}
idx_to_char = {i: c for i, c in enumerate(chars)}

encoded = np.array([char_to_idx[c] for c in text])

# ================= CREATE SEQUENCES =================
seq_len = 50
X, y = [], []

for i in range(len(encoded) - seq_len):
    X.append(encoded[i:i + seq_len])
    y.append(encoded[i + seq_len])

X = np.array(X)
y = np.array(y)

# ================= GRU MODEL =================
model_gru = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, 64, input_length=seq_len),
    tf.keras.layers.GRU(128),
    tf.keras.layers.Dense(vocab_size, activation="softmax")
])

model_gru.compile(
    loss="sparse_categorical_crossentropy",
    optimizer="adam"
)

# ================= TRAIN =================
```

```
model_gru.fit(X, y, epochs=30, batch_size=64)

# ================= TEMPERATURE SAMPLING =================
def sample_with_temperature(preds, temperature=0.8):
    preds = np.asarray(preds).astype("float64")
    preds = np.log(preds + 1e-8) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    return np.random.choice(len(preds), p=preds)

# ================= TEXT GENERATION =================
def generate_text(seed, length=600, temperature=0.8):
    result = seed

    for _ in range(length):
        encoded_seed = [char_to_idx[c] for c in seed]
        encoded_seed = np.array(encoded_seed).reshape(1, -1)

        preds = model_gru.predict(encoded_seed, verbose=0)[0]
        next_idx = sample_with_temperature(preds, temperature)
        next_char = idx_to_char[next_idx]

        result += next_char
        seed = seed[1:] + next_char

    return result

# ================= OUTPUT =================
seed_text = text[:50]

print("\nGenerated Text (GRU):\n")
print(generate_text(seed_text))
```

```
Epoch 7/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 2.3794
Epoch 8/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 2.2944
```

```
Epoch 15/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1.8296
Epoch 16/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.7622
Epoch 17/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.6804
Epoch 18/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.6320
Epoch 19/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.5564
Epoch 20/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1.4996
Epoch 21/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.4233
Epoch 22/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.3849
Epoch 23/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.3221
Epoch 24/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 1.2244
Epoch 25/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 1.1436
Epoch 26/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.0906
Epoch 27/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1.0658
Epoch 28/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.9607
Epoch 29/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.8888
Epoch 30/30
34/34 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.8382

Generated Text (GRU):


artificial intelligence is one of the most important fo purformation, despente procreas, and vexptien coment of is eupovering is altiorning are
ssed and learning.
upes reatorm the purespere such av computerns, at ernical and creas neural caterons to the thucems require latween abviwion.

ationss computed learnication, and explicealizaty in the thertorn espersial intelligence as edesition, and confificial intelligence encti-le cre
```