Generiranje_recepata_pomoću_GPT_2

June 29, 2023

Autor: Mirna Ladnjak, Odjel za matematiku, Osijek

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
import torch
USE_CUDA = torch.cuda.is_available()
device = torch.device('cuda:0') if USE_CUDA else torch.device('cpu')
device
```

[ ]:
```
device(type='cuda', index=0)
```

[ ]:
```
import os
os.chdir("drive/My Drive/NLP_projekt")
```

[ ]:
```
import pandas as pd
df = pd.read_csv('recipes_dataset_1000.csv')
```

[ ]:
```
df.shape
```

[ ]:
```
(1000, 6)
```

[ ]:
```
df
```

[ ]:
```
     Unnamed: 0.1  Unnamed: 0  \
0               0     2150000
1               1     2150001
2               2     2150002
3               3     2150003
4               4     2150004
..            ...         ...
995           995     2150995
996           996     2150996
997           997     2150997
998           998     2150998
999           999     2150999
```

```
                                                  title  \
0                           Michigan Sauce Southern Style
1                         Alsatian Stuffed Chicken Breasts
2                                              Potato Bake
3    Fettuccine With Summer Vegetables and Goat Cheese
4                                Don Pablo's Fresh Salsa
..                                                   ...
995          How to Roast a Pumpkin and Make Puree 101
996                            Turkey Breast "Porchetta"
997                 Onion, Sage and Mozzarella Focaccia
998                                Pumpkin Pear Waffles
999          White Chocolate Rum and Raisin Cheesecake

                                              ingredients  \
0    ["2 lbs extra lean ground chuck (may substitut...
1    ["1 lb chicken breast", "18 lb deli ham, cut i...
2    ["4 cups mashed potatoes", "12 cup sauteed mus...
3    ["Kosher salt", "1 large yellow tomato, seeded...
4    ["1 2/3 can diced tomatoes", "1 medium red or ...
..                                                   ...
995  ["1 sugar pumpkin (2-4 pounds, or as many pump...
996  ["2 teaspoons fennel seed", "2 teaspoons orang...
997  ["1 each Focaccia Dough, prepared (click to vi...
998  ["1 cup flour, whole-wheat pastry", "1 12 teas...
999  ["7 oz. digestive biscuits", "2-3/4 oz. butter...

                                               directions  \
0    ["Put all ingredients into a pot and mix toget...
1    ["Saute ham, shallots and mushrooms together i...
2    ["Blend ingredients together, keeping some of ...
3    ["Bring a large pot of salted water to a boil...
4    ["Place tomatoes, onions, tomato paste, water,...
..                                                   ...
995  ["Roasting 101:.", "Preheat the oven to 350F a...
996  ["For the porchetta: The day before roasting, ...
997  ["Preheat the oven to 450 degrees.", "Once the...
998  ["Mix together the flour, baking powder, cinna...
999  ["Pre-heat oven to 180C.", "(350F.).", "Blend ...

                                                   NER
0    ["extra lean ground chuck", "ketchup", "onion"...
1    ["chicken breast", "deli ham", "shallots", "ba...
2    ["potatoes", "mushroom", "onion", "paprika", "...
3    ["Kosher salt", "yellow tomato", "yellow squas...
4    ["tomatoes", "red", "tomato paste", "water", "...
..                                                   ...
```

```
995                                  ["sugar", "oil"]
996  ["fennel seed", "orange zest", "kosher salt", ...
997  ["Focaccia", "mozzarella cheese", "onions", "o...
998  ["flour", "baking powder", "cinnamon", "nutmeg...
999  ["digestive biscuits", "butter", "dark raisins...

[1000 rows x 6 columns]
```

[ ]:
```
import re
```

[ ]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

[ ]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

[ ]:
```
True
```

[ ]:
```
def preprocess_function(line):
  l1 = line.lstrip('["').rstrip('"]')
  l2 = l1.split(".", "')
  dir = []
  for r in l2:
    new_string = re.sub(r'[^\w\s]', '', r)
    r = new_string
    dir.append(r)

    dir2 = []
    for sent in dir:
      #print(sent)
      word_tokens = word_tokenize(sent)
      filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]

      filtered_sentence = []

      for w in word_tokens:
          if w not in stop_words:
              filtered_sentence.append(w)
```

```
        linee = ' '.join(str(e) for e in filtered_sentence)
        dir2.append(linee)
    #print(dir2)

    return dir2
```

[ ]:
```
def preprocess_function_ing(line):
    l1 = line.lstrip('["').rstrip('"]')
    l2 = l1.split('", "')
    dir = []
    for r in l2:
      new_string = re.sub(r'[^\w\s]', '', r)
      r = new_string
      dir.append(r)

      dir2 = []
      for sent in dir:
        #print(sent)
        word_tokens = word_tokenize(sent)
        filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]

        filtered_sentence = []

        for w in word_tokens:
            if w not in stop_words:
                filtered_sentence.append(w)

        linee = ' '.join(str(e) for e in filtered_sentence)
        dir2.append(linee)
    #print(dir2)



    return dir2
```

[ ]:
```
result = []
recipe=[]

for index, row in df.iterrows():
    processed_title = row["title"]
    processed_ingredients = preprocess_function_ing(row["NER"])
    processed_directions = preprocess_function(row["directions"])

    recipe = "name: " + processed_title + "\ndirections: " + ('. '.
 ↪join(processed_directions)) + "\ningredients: " + (', '.
 ↪join(processed_ingredients))
```

```
  result.append(recipe)
```

[ ]:
```
result[:3]
```

[ ]:
```
['name: Michigan Sauce Southern Style\ndirections: Put ingredients pot mix
together well cooks medium heat. The meat ketchup mixed prior cooking prevents
meat clumping gives best chili texture. This also cooked crock pot\ningredients:
extra lean ground chuck, ketchup, onion, garlic, cumin, chili powder, cayenne
pepper, salt',
 'name: Alsatian Stuffed Chicken Breasts\ndirections: Saute ham shallots
mushrooms together oil. Slit pocket chicken breast. Divide ham mixture evenly
among breasts. Bake 375 covered dish 25 minutes cooked. Remove cover top chicken
shredded cheese. Broil cheese bubbles browns 5 minutes\ningredients: chicken
breast, deli ham, shallots, baby portabella mushrooms, olive oil, shredded
gruyere',
 'name: Potato Bake\ndirections: Blend ingredients together keeping cheddar
aside. Pour lightly oiled baking dish sprinkle top remaining cheese. Bake hot 30
minutes 350 degrees F. To brown cheese broil minutes end cooking\ningredients:
potatoes, mushroom, onion, paprika, mustard powder, basil, garlic, yogurt, egg,
cheddar cheese']
```

[ ]:
```
len(result)
```

[ ]:
```
1000
```

[ ]:
```
#Sort the list of data
result.sort(key = len)
```

[ ]:
```
result = result[:100]
```

[ ]:
```
max = len(result[0])
for i in range(len(result)):
  if len(result[i]) > max:
    print(i)
    print(len(result[i]))
    max = len(result[i])
print(max)
```

```
1
92
2
109
3
112
4
117
...
92
```

```
246
93
247
96
248
97
250
99
253
253
```

[1]:
```
pip install transformers[torch]
```

```
Collecting transformers[torch]
  Downloading transformers-4.30.2-py3-none-any.whl (7.2 MB)


  7.2/7.2 MB
47.0 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers[torch]) (3.12.2)
Collecting huggingface-hub<1.0,>=0.14.1 (from transformers[torch])
  Downloading huggingface_hub-0.15.1-py3-none-any.whl (236 kB)


  236.8/236.8 kB
23.4 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch]) (1.22.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch]) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers[torch]) (6.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2022.10.31)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from transformers[torch]) (2.27.1)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1 (from transformers[torch])
  Downloading
tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(7.8 MB)


  7.8/7.8 MB
67.5 MB/s eta 0:00:00
Collecting safetensors>=0.3.1 (from transformers[torch])
  Downloading
```

```
safetensors-0.3.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)


  1.3/1.3 MB
72.7 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch]) (4.65.0)
Requirement already satisfied: torch!=1.12.0,>=1.9 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2.0.1+cu118)
Collecting accelerate>=0.20.2 (from transformers[torch])
  Downloading accelerate-0.20.3-py3-none-any.whl (227 kB)


 227.6/227.6 kB
16.0 MB/s eta 0:00:00
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-
packages (from accelerate>=0.20.2->transformers[torch]) (5.9.5)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub<1.0,>=0.14.1->transformers[torch]) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.14.1->transformers[torch]) (4.6.3)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages
(from torch!=1.12.0,>=1.9->transformers[torch]) (1.11.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch!=1.12.0,>=1.9->transformers[torch]) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch!=1.12.0,>=1.9->transformers[torch]) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-
packages (from torch!=1.12.0,>=1.9->transformers[torch]) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch!=1.12.0,>=1.9->transformers[torch]) (3.25.2)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch!=1.12.0,>=1.9->transformers[torch]) (16.0.6)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers[torch])
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers[torch])
(2023.5.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers[torch])
(2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->transformers[torch]) (3.4)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from
```

```
jinja2->torch!=1.12.0,>=1.9->transformers[torch]) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-
packages (from sympy->torch!=1.12.0,>=1.9->transformers[torch]) (1.3.0)
Installing collected packages: tokenizers, safetensors, huggingface-hub,
transformers, accelerate
Successfully installed accelerate-0.20.3 huggingface-hub-0.15.1
safetensors-0.3.1 tokenizers-0.13.3 transformers-4.30.2
```

[ ]:
```
import torch
from torch.utils.data import DataLoader, TensorDataset
from sklearn.model_selection import train_test_split
from transformers import GPT2Tokenizer
```

[ ]:
```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
```

```
Downloading (...)olve/main/vocab.json:   0%|          | 0.00/1.04M [00:00<?, ?B/s]
Downloading (...)olve/main/merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
Downloading (...)lve/main/config.json:   0%|          | 0.00/665 [00:00<?, ?B/s]
```

[ ]:
```
# Tokenize and encode the text data
encoded_data = [tokenizer.encode(text) for text in result]

max_sequence_length = 150
truncated_sequences = [sequence[:max_sequence_length] for sequence in␣
 ↪encoded_data]

# Pad the sequences to have equal length
padded_data = [sequence + [0] * (max_sequence_length - len(sequence)) for␣
 ↪sequence in truncated_sequences]
```

[ ]:
```
dec_data = [tokenizer.decode(text) for text in encoded_data]
```

[ ]:
```
for sequence in dec_data[0:50]:
  print(sequence[:150])
```

```
name: Amaretto Cooler Recipe
directions: Ice rocks
ingredients: orange juice, sprite
name: Cocoa Dream Cookies Recipe
directions: Preheat oven 375
ingredients: allpurpose, Nestl
name: Banana Milkshake Recipe
directions: Blend blender till smooth. Serve
ingredients: Banana, milk, Vanilla
name: Indulgence
directions: Layer following order Dark Cacao Amaretto Amarula
ingredients: Cacao, cream liqueur
name: Hummus
```

directions: blend
ingredients: garbanzo beans, tahini, extra virgin olive oil, lemon juice,
garlic, salt
name: Pasta Seafood Salad
directions: Toss together enjoy
ingredients: pasta, frozen peas, red onion, dressing, crabmeat
name: Herb Butter Recipe
directions: Mix well serve bread
ingredients: chives, dill, garlic salt, oregano, parsley flakes, butter
name: chicken tacos
directions: add together let cook 8 hours. Then serve soft shell joy
ingredients: chicken breast, chicken broth
name: Curry Vegetable Dip Recipe
directions: Mix together chill
ingredients: Miracle, horseradish, curry pwdr, vinegar, garlic salt, onion
name: The Bellini
directions: In champagne flute add peach puree top glass chilled Prosecco
ingredients: Italian sparkling wine, peach puree
name: Nutty Hot Chocolate
directions: Combine ingredients large mug mix well
ingredients: chocolate syrup, coconut syrup, almond syrup, milk
name: Southwestern Fiesta Topper
directions: Top crackers remaining ingredients
ingredients: Crackers, Cheddar Cheese, light sour cream, salsa
name: Spiced Ketchup Burger Topping
directions: Mix ingredients together small bowl. Serve chilled
ingredients: ketchup, horseradish, hot sauce
name: Lacama (Spice)
directions: Mix spices together keep airtight box
ingredients: cinnamon, black pepper, curry powder, ginger ground, nutmeg
name: Vegan Mocha Cream Sauce
directions: Blend ingredients together smooth creamy. Swirl desserts
ingredients: tofu, maple syrup, coffee, vanilla
name: The Vijay Singh
directions: Pour syrup gin Chartreuse lemon juice ice. Stir top tonic
ingredients: green tea syrup, gin, Chartreuse, lemon juice
name: Berry Smoothies
directions: Combine ingredients blender process smooth
ingredients: yogurt, nonfat milk, fresh blue berries, banana, chopped ice
name: Basic Foccacia
directions: Combine ingredients bread machine press Start
ingredients: water, sugar, yeast, flour, salt, olive oil, basil, parmes
name: Nutella Hot Chocolate
directions: Boil milk pot Add Nutella preferred sweetness Stir Nutella blended
milk Pour cup serve
ingredients: Milk, Nute
name: Vegetable-Fruit Platter
directions: Arrange fruit vegetables platter. Sprinkle Hunza salt

ingredients: tomatoes, cucumber, avocado, pineapple, s
name: Arawak Recipe
directions: Shake well cracked ice. Strain old fashioned glass filled ice
ingredients: Pineapple Juice, Cranberry Juice, Lime Juic
name: Blue Chimney Smoke
directions: Place ice large wineglass. Add tequila orange juice stir. Float
syrup top
ingredients: curacao syrup, orange juic
name: Cure for Sugar Craving Fudge
directions: Mix together smooth. You also mix Natural Peanut Butter. Enjoy
ingredients: cocoa, Splenda sugar substi
name: Sugar Free Vanilla Chill (South Beach Diet Friendly)
directions: Blend frothy
ingredients: nonfat plain yogurt, soymilk, vanilla, handful ice cu
name: Ranch Ball Recipe
directions: Combine Ranch Mix cream cheese bacon bits. Form ball Chill. Serve
various types crackers
ingredients: cream cheese
name: Magnolia's Alcoholic Beverage Recipe
directions: Mix ingredients. Pour glasses ice garnish oranges cherries
ingredients: orange juice, champagne
name: Dreamsicle
directions: Combine creme de noyaux orange juice halfandhalf ice cocktail
shaker. Shake strain cocktail glass serve
ingredients: noya
name: Mexican Gin and Tonic
directions: Fill highball glass ice. Add gin fresh lime juice. Top glass tonic
stir gently. Garnish lime slice
ingredients
name: Dijon Vinaigrette
directions: Combine ingredients eat fruity nutty otherwise delicious salad
ingredients: olive oil, balsamic vinegar, mustard,
name: Lemon Marinade
directions: Mix ingredients together large bowl. Use required
ingredients: lemon, lemons, olive oil, thyme, sage, scallions, fres
name: Chef's Seasoned Salt
directions: Mix ingredients small bowl store shaker
ingredients: kosher salt, ground white pepper, garlic, ground sundried
name: The Witchs Eye
directions: In shaker muddle grapes. Add Pisco strain flute. Top sparkling sake.
Garnish frozen whole grape eye
ingredients: fres
name: Light Sweet Cream Base for Ice Cream
directions: In mixing bowl whisk light cream sweetened condensed milk together
blended. Makes 1 quart
ingre
name: Corn Casserole
directions: Combine ingredients. Pour greased 9x13 inch dish. Bake 350 degrees 1

hour
ingredients: whole kernel corn, corn, corn
name: Chili-Parmesan Nut Snack Mix
directions: Toss popcorn Parmesan cheese chili powder. Add pretzels nuts mix lightly
ingredients: popcorn, Parmesan
name: Super Easy Black Bean, Corn, and Zucchini Salsa
directions: Mix. Refrigerate. Enjoy
ingredients: frozen corn, black beans, zucchini, cilantro, s
name: Crock Pot Corn Pudding
directions: Combine corn. Pour crock pot. Add corn mix well. Cover cook low 4 hours
ingredients: eggs, sugar, salt, peppe
name: Golden Topped Eggs Recipe
directions: Blend soup lowfat milk. Heat. Top slice toast ham slice poached egg. Pour cheese mix
ingredients: Cheddar
name: breakfast torilla
directions: To updated
ingredients: tobasco sauce, butter, ground black pepper, garlic powder, onion powder, white rice, ameri
name: Chicken, Pear & Asiago Sandwich
directions: Spread bread slices mayo. Fill remaining ingredients
ingredients: multigrain bread, Mayonnaise, Chic
name: Garlic Flavored Oil
directions: Combine ingredients. Put container tight fitting lid store room temperature 6 months
ingredients: olive oil, ros
name: Corn Crab Cakes
directions: Mash together. Fry side well
ingredients: crabmeat, egg, corn, hot sauce, lemon juice, mustard, pepper, nonfat sour
name: Breakfast Cocktail
directions: Combine ingredients pitcher refrigerate serving. Garnish glass slice fresh orange
ingredients: freshly squeezed o
name: Fruity Popcorn Medley
directions: Cut apple peach small chunks. Mix fruit popcorn ad honey mix everything coated add honey needed. Enjoy
ingredi
name: Cranberry Margaritas
directions: Combine ingredients blender blend high. Serve. If blender small highpowered make margaritas 2 batches
ingredien
name: pork or rib sauce
directions: mix ingredients together pour pork ribs bake
ingredients: water, worstshire sauce, brown sugar, paprika, ketchup,
name: Honey Roasted Chicken Salad

```
directions: Toss salad greens chicken vegetables nuts large bowl. Add dressing
mix lightly
ingredients: salad greens
name: Taco Quarter Pounders
directions: Mix together shape 6 patties. Grill fry desired doneness. Serve
toasted buns favorite taco toppings side
ingre
name: Tamale Balls Recipe
directions: Combine mix well. Form small balls. Place following sauce 2 hrs
ingredients: beef, pork sausage, corn meal, toma
name: Cabbage Slaw
directions: Combine cabbage tomato green onion cilantro. Use judgment lime
juice. Slaw better longer sits
ingredients: green cabbag
```

[ ]:
```
# Convert the padded sequences to a tensor
input_ids = torch.tensor(padded_data)
input_ids.shape
```

[ ]:
```
torch.Size([100, 150])
```

[ ]:
```
target_ids = input_ids
```

[ ]:
```
# Split the data into training and validation sets
train_inputs, val_inputs, train_targets, val_targets =␣
 ↪train_test_split(input_ids, target_ids, test_size=0.2, random_state=42)

# Create TensorDatasets for training and validation
train_dataset = TensorDataset(train_inputs, train_targets)
val_dataset = TensorDataset(val_inputs, val_targets)
train_dataset
```

[ ]:
```
<torch.utils.data.dataset.TensorDataset at 0x7fbaf580e380>
```

[ ]:
```
train_inputs.shape
```

[ ]:
```
torch.Size([80, 150])
```

[ ]:
```
val_inputs.shape
```

[ ]:
```
torch.Size([20, 150])
```

[ ]:
```
# Create DataLoaders for training and validation
batch_size = 4
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
```

[ ]:
```
from transformers import GPT2LMHeadModel
```

[ ]:
```
# Load the GPT-2 model
model = GPT2LMHeadModel.from_pretrained('gpt2')

# Define the optimizer
optimizer = torch.optim.Adam(model.parameters(), lr=1e-4)
```

```
Downloading model.safetensors:   0%|          | 0.00/548M [00:00<?, ?B/s]
```

```
Downloading (...)neration_config.json:   0%|          | 0.00/124 [00:00<?, ?B/s]
```

[ ]:
```
def rate(step, model_size, factor, warmup):
    if step == 0:
        step = 1
    return factor * (model_size ** (-0.5) * min(step ** (-0.5), step * warmup **␣
 ↪(-1.5)))

num_epochs = 10
output_dir = './results'

model_size = 117
factor = 1.0
warmup = 1000
```

[ ]:
```
for epoch in range(num_epochs):
    model.train()
    for step, (input_ids, target_ids) in enumerate(train_loader):
        optimizer.zero_grad()
        outputs = model(input_ids=input_ids, labels=target_ids)
        loss = outputs.loss
        loss.backward()
        optimizer.step()

        # Adjust learning rate
        lr = rate(step, model_size, factor, warmup)
        for param_group in optimizer.param_groups:
            param_group['lr'] = lr

    # Save the model
    model_save_path = f"{output_dir}/epoch_{epoch+1}.pt"
    torch.save(model.state_dict(), model_save_path)

    # Validation loop
    model.eval()
    val_loss = 0.0
    with torch.no_grad():
        for input_ids, target_ids in val_loader:
            outputs = model(input_ids=input_ids, labels=target_ids)
            val_loss += outputs.loss.item()
```

```
    print(f"Epoch {epoch+1}: Validation Loss = {val_loss/len(val_loader):.4f}")

# Save the fine-tuned model
model.save_pretrained('fine-tuned-model')
```

```
Epoch 1: Validation Loss = 1.0311
Epoch 2: Validation Loss = 0.9357
Epoch 3: Validation Loss = 0.9215
Epoch 4: Validation Loss = 0.9204
Epoch 5: Validation Loss = 0.9250
Epoch 6: Validation Loss = 0.9363
Epoch 7: Validation Loss = 0.9489
Epoch 8: Validation Loss = 0.9688
Epoch 9: Validation Loss = 0.9827
Epoch 10: Validation Loss = 1.0024
```

[ ]:
```
model = GPT2LMHeadModel.from_pretrained('fine-tuned-model')
```

[ ]:
```
#model = GPT2LMHeadModel.from_pretrained('gpt2')
#model.load_state_dict(torch.load('./results/epoch_5.pt'))
```

[ ]:
```
model.eval()
# Move the model to the device
model.to(device)
```

[ ]:
```
GPT2LMHeadModel(
  (transformer): GPT2Model(
    (wte): Embedding(50257, 768)
    (wpe): Embedding(1024, 768)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (0-11): 12 x GPT2Block(
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D()
          (c_proj): Conv1D()
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D()
          (c_proj): Conv1D()
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
```

```
    )
    (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  )
  (lm_head): Linear(in_features=768, out_features=50257, bias=False)
)
```

[ ]:
```python
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')

def generate_recipe(model, tokenizer, max_length=100, temperature=0.7):
    prompt = "New Recipe:"
    input_ids = tokenizer.encode(prompt, return_tensors='pt').to(device)
    output = model.generate(
        input_ids=input_ids,
        max_length=max_length,
        temperature=temperature,
        pad_token_id=tokenizer.eos_token_id,
        num_return_sequences=1
    )
    #print(output)
    generated_recipe = tokenizer.decode(output[0], skip_special_tokens=True)
    return generated_recipe

# Generate a new recipe
new_recipe = generate_recipe(model, tokenizer)
print(new_recipe)
```

```
New Recipe: Cabbage Salad
directions: Combine cabbage tomatoes dressing mix well. Add salt pepper. Mix
well. Serve salad
ingredients: cabbage, tomatoes, salt, pepper, cumin, garlic, cumin, cumin,
cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin,
cumin, cumin, cumin, cumin, cumin, cumin,
```

[2]:
```python
import nltk
```

[14]:
```python
reference_texts = ['name: Michigan Sauce Southern Style\ndirections: Put␣
 ↪ingredients pot mix together well cooks medium heat. The meat ketchup mixed␣
 ↪prior cooking prevents meat clumping gives best chili texture. This also␣
 ↪cooked crock pot\ningredients: extra lean ground chuck, ketchup, onion,␣
 ↪garlic, cumin, chili powder, cayenne pepper, salt']
```

[15]:
```python
generated_texts = ['New Recipe: Cabbage Salad \ndirections: Combine cabbage␣
 ↪tomatoes dressing mix well. Add salt pepper. Mix well. Serve salad␣
 ↪\ningredients: cabbage, tomatoes, salt, pepper, cumin, garlic, cumin, cumin,␣
 ↪cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin, cumin,␣
 ↪cumin, cumin, cumin, cumin, cumin, cumin']
```

[16]:
```
# Convert generated and reference texts into tokenized lists
generated_tokens = [generated_text.split() for generated_text in generated_texts]
reference_tokens = [[reference_text.split()] for reference_text in␣
 ↪reference_texts]
```

[17]:
```
# Compute BLEU score
bleu_score = nltk.translate.bleu_score.corpus_bleu(reference_tokens,␣
 ↪generated_tokens)
bleu_score
```

[17]: 3.4364138069410433e-155

[ ]:

0.1 Different results

[ ]:
```
from IPython.display import Image
```

Example 1: num_data = 10, batch_size = 4

```
New Recipe: Grilled Squid with Corn, Tomatoes, and Freshly Tomatoes
ingredients: corn, tomato paste, extra virgin olive oil, unsalted butter, light brown sugar, Kosher salt, egg, baby portabella
mushrooms, shallots, shallots, shallots with feta cheese
directions: Bring a large pot of salted water to a boil. Add the corn and tomato paste and cook stirring occasionally until the
liquid separates from the liquid. Add the tomato paste
```

Example 2: num_data = 100, max_sequence_length = 100, batch_size = 8

```
New Recipe: Chicken Stock
directions: In large bowl combine chicken stock, salt, pepper, garlic, onion,
cumin, cumin powder, salt, pepper, cumin powder, cayenne pepper. Stir well.
Cover chicken stock tightly. Cover chicken wire rack. Place chicken wire rack.
Place chicken wire rack. Place chicken wire rack. Place chicken wire rack. Place
chicken wire rack. Place chicken wire rack. Place chicken wire rack. Place
chicken wire rack. Place chicken wire
```

Example 3: num_data = 100, max_sequence_length = 100, batch_size = 16

```
print(new_recipe)
```

```
New Recipe:
Ingredients:
1 cup flour
1 cup sugar
1 cup water
1 cup milk
1 teaspoon salt
1 teaspoon baking powder
1 teaspoon baking soda
1 teaspoon baking powder
1 teaspoon baking soda
1 teaspoon baking powder
1 teaspoon baking soda
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking powder
1 teaspoon baking
```

Example 4: num_data = 1000, max_sequence_length = 100, batch_size = 8

```
New Recipe: Sweet Potato Salad
ingredients: potatoes, carrots, celery, onion, green chile, garlic, salt,
pepper, cumin, cumin powder, cumin powder, cumin powder, cumin powder, cumin
```

Example 5: num_data = 1000, max_sequence_length = 300, batch_size = 4

```
New Recipe: Chicken Casserole
directions: Mix all ingredients together. Bake at 350u00b0 for 45 minutes
ingredients: chicken, onion, garlic, salt, pepper, chicken broth, chicken
bouillon cubes, chicken bouillon cubes, chicken bouillon cubes, chicken bouillon
cubes, chicken bouillon cubes, chicken bouillon cubes, chicken bouillon cubes,
chicken bouillon cubes, chicken bouillon cubes, chicken bouillon cubes, chicken
bouillon cubes, chicken bou
```

```
[ ]:
```