```
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch


model_name = "gpt2"
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%                                  26.0/26.0 [00:00<00:00, 2.34kB/s]
vocab.json: 100%                                         1.04M/1.04M [00:00<00:00, 7.16MB/s]
merges.txt: 100%                                         456k/456k [00:00<00:00, 21.2MB/s]
tokenizer.json: 100%                                       1.36M/1.36M [00:00<00:00, 18.5MB/s]
config.json: 100%                                        665/665 [00:00<00:00, 38.7kB/s]
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better perfo
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to r
model.safetensors: 100%                                    548M/548M [00:06<00:00, 158MB/s]
generation_config.json: 100%                                   124/124 [00:00<00:00, 10.0kB/s]
```

```
model.eval()
```

```
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(nf=2304, nx=768)
          (c_proj): Conv1D(nf=768, nx=768)
          (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(nf=3072, nx=768)
          (c_proj): Conv1D(nf=768, nx=3072)
          (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)
)
```

```
prompt = "Once upon a time in the world of Artificial Intelligence,"
input_ids = tokenizer.encode(prompt, return_tensors="pt")


with torch.no_grad():  # Disable gradient calculation
    output_ids = model.generate(
        input_ids,
        max_length=100,
        num_return_sequences=1,
        temperature=0.7,  # Controls creativity; lower = more focused
        top_k=50,         # Limits to top 50 likely next words
        top_p=0.95,       # Nucleus sampling (top-p sampling)
        do_sample=True    # Enables sampling instead of greedy search
    )
```

```
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `a
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
The attention mask is not set and cannot be inferred from input because pad token is same as eos token. As a consequence, you may observ
```

```
generated_text = tokenizer.decode(output_ids[0], skip_special_tokens=True)
print(generated_text)
```

```
Once upon a time in the world of Artificial Intelligence, we saw the beginning of an era of artificial intelligence. One of the most int

The Human Machine was created by the Artificial Intelligence of the United States, in order to train humanity in its natural and human q
```

Start coding or generate with AI.