

IMPORT

```
from transformers import GPT2Tokenizer, GPT2LMHeadModel  
import torch
```

Saya memulai program ini dengan mengimpor dua komponen utama dari library Transformers, yaitu GPT2Tokenizer dan GPT2LMHeadModel, serta library PyTorch (torch). Tokenizer berfungsi untuk mengubah teks menjadi token numerik yang bisa dipahami oleh model, sedangkan GPT2LMHeadModel adalah model bahasa GPT-2 yang sudah dilatih untuk menghasilkan teks.

Set up tokenizer dan Model

```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")  
model = GPT2LMHeadModel.from_pretrained("gpt2")  
  
/usr/local/lib/python3.12/dist-packages/huggingface_hub/  
file_download.py:945: FutureWarning: `resume_download` is deprecated  
and will be removed in version 1.0.0. Downloads always resume when  
possible. If you want to force a new download, use  
`force_download=True`.  
warnings.warn(  
  
{"model_id": "06656778775246dfb92463dd5d697c53", "version_major": 2, "vers  
ion_minor": 0}  
  
{"model_id": "164a6d32ea3f4000a298df86585d5604", "version_major": 2, "vers  
ion_minor": 0}
```

Kemudian, saya memuat model dan tokenizer GPT-2 versi standar yang sudah tersedia dari Hugging Face Hub. Perintah ini secara otomatis mengunduh model pretrained dan konfigurasi tokenizernya.

Tokenisasi

```
blog_title = "Weekend Gateway"
```

Selanjutnya, saya membuat sebuah variabel blog_title yang berisi teks awal atau prompt untuk model, yaitu "Weekend Gateway". Teks ini akan menjadi titik awal bagi GPT-2 untuk melanjutkan menulis kalimat berikutnya.

```
input_ids = tokenizer.encode(blog_title, return_tensors='pt')
```

Lalu, saya mengubah teks tersebut menjadi representasi token numerik menggunakan tokenizer GPT-2. Karena GPT-2 hanya dapat memproses angka, hasil encoding ini akan dikembalikan dalam bentuk tensor PyTorch melalui parameter `return_tensors='pt'`.

```
input_ids  
tensor([[20916,    437, 29916]])
```

Variabel `input_ids` sekarang berisi token numerik yang mewakili kata "Weekend Gateway". Saya bisa melihat isi tensor ini hanya dengan memanggil `input_ids`.

Generate

```
output_ids = model.generate(  
    input_ids,  
    max_length=100,  
    num_beams=5,  
    no_repeat_ngram_size=2,  
    early_stopping=True  
)
```

The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results.
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

Berikutnya, saya menggunakan metode `generate()` dari model GPT-2 untuk membuat teks lanjutan berdasarkan prompt tersebut. Saya menambahkan beberapa parameter agar hasilnya lebih rapi dan natural :

- `max_length=100` berarti panjang maksimal teks yang dihasilkan adalah 100 token.
- `num_beams=5` artinya saya menggunakan teknik beam search dengan 5 jalur kemungkinan, agar hasilnya lebih berkualitas daripada sekadar acak.
- `no_repeat_ngram_size=2` memastikan model tidak mengulang frasa dua kata yang sama secara berurutan.
- `early_stopping=True` membuat proses berhenti lebih cepat jika semua kemungkinan hasil sudah selesai dipertimbangkan.

```
# Decode hasilnya kembali ke teks  
generated_text = tokenizer.decode(output_ids[0],  
skip_special_tokens=True)  
  
print(generated_text)
```

Weekend Gateway

This is where you'll find all of the information you need to get started. It's a great place to start if you're looking to learn more

about how to build a website. If you have any questions, feel free to reach out to me on Twitter or email me at

Setelah model menghasilkan output berupa token numerik baru, saya perlu mengubahnya kembali menjadi teks yang bisa dibaca manusia. Untuk itu saya menggunakan fungsi decode() dari tokenizer.

Terakhir, saya menampilkan hasil teks yang dihasilkan oleh GPT-2 menggunakan fungsi print().