

RAHMA PUTRI P

11122170

PRAKTIKUM ROBOTIKA

RAHMA PUTRI_Pertemuan 5 - Moda Hands-On Automated Article Generation.ipynb

Dalam praktikum ini Anda akan mencoba untuk membuat sebuah artikel dengan library yang telah di training sebelumnya atau yang dikenal sebagai pre-trained model. Apabila Anda akan mengunduh file praktikum dan menjalankan pada mesin local Anda, silahkan pastikan dependency library lainnya telah terinstall. Pada praktikum ini Anda akan menggunakan model GPT2 dari OpenAI.

```
[1] 41s
import tensorflow as tf
from transformers import GPT2LMHeadModel, GPT2Tokenizer
```

PENJELASAN

Konsep Umum: Automated Article Generation

Tujuan utama kode ini: Membuat sistem pembuatan teks/artikel otomatis dengan memanfaatkan model GPT-2 (Generative Pretrained Transformer 2) dari pustaka Hugging Face Transformers.

GPT-2 adalah model language generation berbasis Transformer yang dilatih dengan miliaran teks untuk bisa:

- Memahami konteks kalimat.
- Meneruskan teks secara logis dan alami.
- Menulis artikel, ringkasan, caption, cerita, dll secara otomatis.

Import Library

- TensorFlow (tf): Framework machine learning yang digunakan untuk menjalankan model deep learning.
- Transformers (dari Hugging Face): Library yang menyediakan berbagai model NLP siap pakai seperti BERT, GPT, T5, dan lain-lain.
- GPT2LMHeadModel: Model GPT-2 yang siap digunakan untuk language modeling (yaitu menghasilkan teks).
- GPT2Tokenizer: Tokenizer khusus GPT-2 yang mengubah teks manusia menjadi angka (token ID) agar bisa diproses oleh model.

Set Up Tokenizer dan Model

Pada praktikum ini karena kita menggunakan model yang telah dilatih sebelumnya, kita perlu mendefinisikan dua komponen sebelum dapat membuat artikel secara otomatis yaitu tipe tokenizer dan model yang akan digunakan

```
[1] 2s
tokenizer = GPT2Tokenizer.from_pretrained("gpt2-large")
/usr/local/lib/python3.12/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 secret in your Google Colab and restart your session.
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, 531B/s]
vocab.json: 100% [██████████] 1.04M/1.04M [00:00<00:00, 10.7MB/s]
merges.txt: 100% [██████████] 459k/459k [00:00<00:00, 10.1MB/s]
tokenizer.json: 100% [██████████] 1.30M/1.30M [00:00<00:00, 15.5MB/s]
config.json: 100% [██████████] 0/0 [00:00<00:00, 70.8KB/s]

[2] 20s
model = GPT2LMHeadModel.from_pretrained("gpt2-large", pad_token_id=tokenizer.eos_token_id)
model.state_dict(): 100% [██████████] 3.25G/3.25G [00:28<00:00, 257MB/s]
generation_config.json: 100% [██████████] 124/124 [00:00<00:00, 7.80kB/s]
```

PENJELASAN

- `from_pretrained("gpt2-large")` Artinya kita mengambil model GPT-2 versi besar yang sudah dilatih sebelumnya oleh OpenAI dan diunggah ke Hugging Face.
- `pad_token_id=tokenizer.eos_token_id` GPT-2 tidak punya token khusus untuk padding, jadi kita set End-of-Sentence (EOS) token sebagai pengganti.

Tujuannya: Agar kita tidak perlu melatih model dari nol — cukup gunakan model yang sudah siap (pretrained).

▼ Proses Tokenisasi

Pada praktikum ini karena kita menggunakan model yang telah di latih sebelumnya, kita perlu mendefinisikan dua komponen sebelum dapat membuat artikel secara otomatis yaitu tipe tokenizer dan model yang akan digunakan. Proses tokenisasi pada dasarnya adalah pemisahan frasa, kalimat, paragraf, atau seluruh dokumen teks menjadi unit yang lebih kecil, seperti kata atau istilah individual. Masing-masing unit yang lebih kecil ini disebut token. Dalam tokenization, unit yang lebih kecil dibuat dengan menempatkan batas kata. Batas kata adalah titik akhir dari sebuah kata dan awal dari kata berikutnya. Token ini dianggap sebagai langkah pertama untuk proses stemming dan lemmatization

```
[4] blog_title = "Weekend Gateway"
[5] input = tokenizer.encode(blog_title, return_tensors='pt')
[6] input
tensor([[20916, 437, 29916]])
```

PENJELASAN

- Tokenisasi = proses mengubah teks menjadi token numerik. Misalnya "Weekend Gateway" → [50256, 1289, 3456] (sekadar contoh).
- return_tensors='pt': Mengubah hasil tokenisasi menjadi format tensor PyTorch (pt) agar bisa diproses oleh model.

Alasan digunakan: Model deep learning hanya bisa memahami angka, bukan teks mentah.

▼ Let's Generate!

Pada tahapan ini Anda sudah siap untuk membuat model pertama Anda

```
[7] output = model.generate(input, max_length=100, num_beams=5, no_repeat_ngram_size=2, early_stopping=True)
      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 observe unexpected behavior. Please pass your input's `attention_ma
[8] print(tokenizer.decode(output[0], skip_special_tokens=True))
      Weekend Gateway
      If you're looking for a place to stay during the weekend, we've got you covered. We have a variety of options for you to choose from, including hotels, motels, inns, and more. Check out the list be
```

PENJELASAAN

- max_length=100 Panjang maksimal teks keluaran (100 token). Bisa diubah sesuai kebutuhan.
- num_beams=5 Menggunakan teknik Beam Search agar hasil teks lebih optimal (tidak acak).
- no_repeat_ngram_size=2 Mencegah model mengulang frasa dua kata yang sama (supaya teks lebih alami).
- early_stopping=True Menghentikan proses generasi lebih awal jika model merasa kalimat sudah selesai.